Optimal Alignment of Structures for Finite and Periodic Systems

Matthew Griffiths,* Samuel P. Niblett, and David J. Wales

Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, United Kingdom

E-mail: mg542@cam.ac.uk

Abstract

Finding the optimal alignment between two structures is important for identifying the minimum root-mean-square distance (RMSD) between them and as a starting point for calculating pathways. Most current algorithms for aligning structures are stochastic, scale exponentially with the size of structure, and the performance can be unreliable. We present two complementary methods for aligning structures corresponding to isolated clusters of atoms and to condensed matter described by a periodic cubic supercell. The first method (Go-PERMDIST), a branch and bound algorithm, locates the global minimum RMSD deterministically in polynomial time. The run time increases for larger RMSDs. The second method (FASTOVERLAP) is a heuristic algorithm that aligns structures by finding the global maximum kernel correlation between them using fast Fourier transforms (FFTs) and fast SO(3) transforms (SOFTs). For periodic systems FASTOVERLAP scales with the square of the number of identical atoms in the system, reliably finds the best alignment between structures that are not too distant, and shows significantly better performance than existing algorithms. The expected run time for Go-PERMDIST is longer than FASTOVERLAP for periodic systems.

For finite clusters, the FASTOVERLAP algorithm is competitive with existing algorithms. The expected run time for Go-PERMDIST to find the global RMSD between two structures deterministically is generally longer than for existing stochastic algorithms. However, with an earlier exit condition, Go-PERMDIST exhibits similar or better performance.

1 Introduction

Quantifying the difference or similarity between two structures is of broad relevance. In a chemical context we may be interested in using measures of structural similarity amongst a set of chemical structures to predict various chemical properties, for example in quantitative structure-activity relationships (QSAR), where statistical models for predicting the chemical and biological activities of new structures are generated from data about known structures.[1] In this field many alignment protocols utilise additional information about the structures, so that the alignment is performed primarily on the chemically active region of the structure.

A related field is that of machine learning of chemical properties, where a variety of approaches have been developed. Here it is of particular importance that the methods effectively capture the structural information in ways that it is easy for the machine learning algorithms to learn from.[2–5]

Quantifying the similarity between structures can allow two configurations to be aligned to match each other as closely as possible. This alignment is particularly useful in discrete path sampling (DPS),[6–8] which identifies pathways and transition states between local minima on the energy landscape. The initial pathway obtained between distant minima may be the union of many individual minimum-transition state-minimum paths,[9] and is likely to require extensive refinement to locate kinetically relevant routes. Substantial gains in efficiency are likely if the end points can be aligned to improve the initial interpolation[10,11] and reduce the corresponding path length.

Optimal alignment for connection purposes usually corresponds to minimising the Eu-

clidean distance between the two end points in $3N$-dimensional configuration space, where N is the number of atoms. However, there are cases where the minimum distance results in incorrect local permutational alignment, producing artificially high barriers if the permutations are not corrected.[10] For large biomolecules a local permutational alignment procedure was introduced to solve this problem,[10] combining translational and orientational degrees of freedom with the shortest augmenting path algorithm[12] for each group of permutable atoms and an adjustable number of atoms from the immediate environment.

The Euclidean distance in configuration space is simply related to the root-mean-square distance (RMSD) by a factor of $\sqrt{N}$, so we can use these quantities interchangeably. RMSD is the most commonly used metric for comparing two different structures. However, the exhaustive, deterministic calculation of the minimal RMSD with respect to translational, rotational and permutational symmetries scales combinatorially with the number of identical atoms in the system.[13] The difficulties associated with using RMSD have led to the development of a wide variety of alternative metrics for quantifying the dissimilarity between structures,[14] as discussed below.

The problem of 3D point set registration in computer vision is analogous to structure alignment in chemical systems. It is used in many different applications, for example in 3D surface reconstruction,[15] alignment of magnetic resonance images (MRI)s and computer aided tomography (CAT) scans,[16] optical character recognition,[17] and range image matching.[18] In computer vision the correspondence between point sets usually does not need to be one-to-one, in contrast to most chemical alignment problems.

In the present contribution we present two different approaches for finding the minimal RMSD, or equivalently the optimal alignment between two structures. The first approach is an heuristic method featuring polynomial complexity with respect to the number of identical atoms in the structures; this is a kernel correlation based method,[19,20] referred to as FASTOVERLAP. The second approach is a branch and bound algorithm based on the *Globally Optimal Iterative Closest Point* (Go-ICP) method[21,22] for deterministically determining

the global minimum RMSD between two structures. The computational complexity scales polynomially with respect to system size and RMSD; this method will be referred to as Go-PERMDIST. Both these approaches can be applied to isolated clusters of atoms and periodic structures.

For periodic systems, FASTOVERLAP reliably and efficiently aligns most structures, except for a small number of distant pairs of configurations, and performs significantly better than existing algorithms.

Defining the RMSD Two structures (or in computer vision terms, point sets), p and q, can each be defined by N atomic coordinates, $\mathbf{R}^p = (\mathbf{r}_1^p, \mathbf{r}_2^p, ..., \mathbf{r}_N^p) \in \mathbb{R}^{3N}$ and $\mathbf{R}^q = (\mathbf{r}_1^q, \mathbf{r}_2^q, ..., \mathbf{r}_N^q) \in \mathbb{R}^{3N}$. The generalised Euclidean distance (norm)

$$|\mathbf{R}^p - \mathbf{R}^q| = \left(\sum_{j=1}^{N} |\mathbf{r}_j^p - \mathbf{r}_j^q|^2 \right)^{1/2}, \tag{1}$$

is not a good metric, because it is not invariant to symmetries of the Hamiltonian. For an isolated cluster in the absence of external fields the energy is invariant to overall translation and rotation, and to permutations of identical atoms. Similarly, for a periodic system, point-group symmetries, permutations, and global translations leave the energy unchanged. The RMSD between two structures is better defined as the minimum of the Euclidean norm with respect to all these symmetries. For an isolated cluster this definition becomes

$$\mathrm{RMSD}(p, q) = \frac{1}{\sqrt{N}} \min_{\mathbf{M},\mathbf{P},\mathbf{D}} |\mathbf{R}^p - \mathbf{P}(\mathbf{R}^q \mathbf{M}^\top - \mathbf{D})|, \tag{2}$$

where $\mathbf{P}$ is a $3N \times 3N$ permutation matrix of the atomic coordinates, $\mathbf{D} \in \mathbb{R}^{3N}$ contains N copies of the global displacement vector, $\mathbf{d} \in \mathbb{R}^3$, and $\mathbf{M} \in \mathbb{R}^{3N \times 3N}$ is a block diagonal matrix, containing N copies of a rotation matrix $\mathbf{m} \in \mathrm{SO}(3)$.[23]

Similarly, for a periodic system we can define

$$\mathrm{RMSD}(p,q) = \frac{1}{\sqrt{N}} \min_{\mathbf{L},\mathbf{P},\mathbf{D},\mathbf{S}} |\mathbf{R}^p - \mathbf{P}(\mathbf{R}^q\mathbf{S} - \mathbf{D} - \mathbf{L})|, \tag{3}$$

where $\mathbf{S} \in \mathbb{R}^{3N\times 3N}$ is a block diagonal matrix containing N copies of a 3×3 matrix corresponding to symmetry operations of the periodic supercell, $\mathbf{L} = (\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_N) = L\mathbf{J} \in \mathbb{R}^{3N}$ is a set of lattice vectors, with $\mathbf{J} \in \mathbb{Z}^{3N}$ and L the length of the unit cell. In the present work we consider a cubic supercell, but the above definition is easily generalised.

Calculating the RMSD is therefore a global optimisation problem, requiring the identification of the relative lattice vectors and/or rotation, permutation and translation that defines the global minimum. Locating this global minimum is equivalent to finding the optimal alignment of two structures, where the total squared displacement between them is minimised. Henceforth, we will refer to finding the minimal RMSD between two structures as aligning them.

1.1 Existing Methods

A variety of methods have been developed to calculate the minimal RMSD. They are either heuristic or are not guaranteed to locate the global minimum RMSD in polynomial time. A variety of algorithms have also been developed in the computer vision literature that attempt to minimise alternative metrics, as discussed below.

1.1.1 Partial Algorithms

Various algorithms have been developed that, in polynomial time, will find the global minimum for one of the symmetries over which we are minimising.

Translational Alignment For an isolated cluster it can easily be shown that the best alignment will always occur when the centres of coordinates coincide, independent of the permutations, rotations and the number of different chemical species present.[13]

This result does not apply to a periodic system, because the centre of coordinates is not well defined, although the average displacement between the two structures must be zero when the distance between them is minimised.

Permutational Alignment If the minimisation is restricted to permutations then the optimal permutation can be found in polynomial time using the Hungarian algorithm,[24] which scales approximately as $O(N^{2.5})$ and the shortest-augmenting path algorithm, which is faster and scales as $O(N^2)$.[12] Both these algorithms are forms of primal-dual methods that perform a simultaneous primal constrained maximisation and dual constrained minimisation, when both problems are satisfied then the optimal solution has been found.[12]

Rotational Alignment Finding the optimal rotational alignment for a fixed permutation has an analytic solution, $O(N)$, and can be achieved using quaternions[25] or Lagrange multipliers.[26]

Lattice Vectors For a given displacement and permutation the lattice vector that minimises the RMSD between the two structures can be found in $O(N)$ operations. For a given permutation, finding the global translation and set of lattice vectors that minimises the RMSD also requires $O(N)$ operations.

Point Group Symmetries The RMSD should also be minimal with respect to the point group symmetries of the periodic structure, which can be enumerated. For isolated structures the inverted structure ($\mathbf{R}^q \rightarrow -\mathbf{R}^q$) may also be relevant.

Unfortunately, iteratively minimising each of these symmetries in turn does not guarantee that the global minimum RMSD will be found. The only way to guarantee this condition is by testing every possible permutation. Since there are $N!$ possible permutations for a homoatomic system this approach is prohibitively expensive for all but the smallest systems.

1.1.2 Full Algorithms

Monte Carlo Alignment Sadeghi et al.[13] developed a Monte Carlo algorithm for calculating the global minimum RMSD for both clusters and periodic systems.[27] In this method an initial permutational alignment is performed by either matching the principal axes of the moment of inertia, or by matching atoms with similar local environments. Random permutations followed by a rotational or displacement alignment are then applied, and the new alignment is accepted if the RMSD is less than the old RMSD plus a small adjustable parameter. This parameter is changed dynamically during the simulation to keep the acceptance rate around 50%. The number of MC iterations required to find the global minimum RMSD for this method scales approximately exponentially for atomic Lennard-Jones clusters.[13]

Iterative Closest Point (ICP) This is one of the most commonly used algorithms in computer vision to align point sets. This algorithm iteratively pairs up the closest points in the two point clouds and then minimises the distance squared between them until no further improvement appears.[18] Variants of this method have been developed, incorporating the expectation-maximisation algorithm[28] or using the Levenberg-Marquardt approach.[17] These local minimisation methods can be combined with a branch and bound scheme to find the global minimum of the cost function.[21,22,29] However, because the cost function measures the nearest neighbour distance, these algorithms will not necessarily find the global RMSD.

Kernel Correlation An alternative approach developed for point set registration is based on maximising the kernel correlation between two points sets, p and q. For a kernel function, $K(\mathbf{r}, \mathbf{r}')$, we can define the kernel correlation between two points, $\mathbf{r}_j^p$ and $\mathbf{r}_{j'}^q$ as,

$$KC(\mathbf{r}_j^p, \mathbf{r}_{j'}^q) = \int K(\mathbf{r}, \mathbf{r}_j^p) K(\mathbf{r}, \mathbf{r}_{j'}^q) \, \mathrm{d}\mathbf{r}, \tag{4}$$

so the total kernel correlation can be calculated as

$$KT(p,q) = \sum_{j} \sum_{j'} KC(\mathbf{r}_j^p, \mathbf{r}_{j'}^q). \tag{5}$$

The registration of two point sets is achieved by performing a non-linear optimisation of the total kernel correlation.[19] This method is directly analogous to the extended Gaussian image (EGI) approach.[30] EGI was used by Makadia et al.[20] to align point sets with very little overlap, optimising rotations with the $SO(3)$ Fourier transform (SOFT)[31] to find the best correlation between discrete histograms of the EGI images. The SOFT has also been used to identify binding regions between proteins.[32,33]

Branch and Bound RMSD Hong et al.[34] developed a branch and bound based method for deterministically calculating the RMSD between two configurations of identical atoms. The algorithm works by progressively bounding the RMSD between subsets of the atoms in both structures. By bounding the lowest possible RMSD for each subset the algorithm can eliminate those that give poorer alignments, removing that region of search space. The algorithm exhibited better than $O(N^2)$ performance for aligning identical but displaced structures of random data.

Our own analysis and implementation of this algorithm suggests that the performance is not competitive for alignment of different structures. The number of permutation subsets with a lower bound below a given distance scales approximately exponentially with the distance, which means that the computational complexity scales approximately with the exponential of the minimum RMSD.

Methods Implemented in Cambridge Energy Landscape Software PERMDIST, ATOMMATCHFULL and ATOMMATCHDIST are heuristic algorithms for estimating the global RMSD, which have been developed and implemented in the public domain programs GMIN[35] and OPTIM.[36] These algorithms are described below:

PERMDIST This algorithm applies a successive set of permutational alignments, using the shortest augmenting path algorithm,[12] each one followed by an overall rotational or translational alignment.[10] The procedure is repeated until a minimum RMSD in permutational space is reached. Because this process is not guaranteed to give the global RMSD it is restarted from multiple random initial rotations/displacements. This approach has much in common with the ICP based algorithms.

ATOMMATCHFULL This algorithm was developed to identify structural isomers of periodic systems by successively superimposing every pair of atoms and then checking how many other atoms in one structure are within a certain distance of an atom in the second structure (in which case the two atoms are said to "match").[37] An exhaustive search is performed, superimposing all pairs of atoms within the smallest permutable group, which allows us to fix the global translation. Once the global translation is found the permutational assignment problem can be solved to get the full permutation. Because this algorithm attempts to maximise the number of matches it does not necessarily find the global RMSD. It scales approximately as $O(N^4)$, so for large systems it is computationally expensive.

ATOMMATCHDIST This algorithm is based on ATOMMATCHFULL, but reduces the computational expense by exiting some of the loops over atoms early if the current trial superposition does not give enough matches.[37] This strategy sometimes gives significantly poorer alignments than ATOMMATCHFULL.

Methods Implemented in KPLOT Two algorithms have been developed for use in the structure visualisation and analysis program KPLOT[38] for identifying isostructural similarities between structures. These methods do not attempt to minimise the RMSD, but are included for reference.

CMPZ This is a method developed for comparing crystallographic structures. It looks for the set of affine transformations that map atoms from the rescaled unit cell of one crystal structure onto atoms in the rescaled unit cell of the second structure; it then performs the inverse transform to check that the mapping is bijective. If all the atoms map to within a certain tolerance of each other then the structures are described as equivalent.[39] If the unit cells have different specifications the algorithm will detect whether they are equivalent, whether one unit cell is a supercell of the other, and/or whether one structure is a substructure of the other.

CCL This approach extends the CMPZ algorithm to clusters, by seeking the affine transformation that maps one set of atoms onto another, and it can be used to identify structural isomers. It will also identify whether a cluster forms a smaller part of a larger system.[40]

1.1.3 Alternative Metrics

Due to several difficulties associated with calculating and using the RMSD a variety of alternative metrics and descriptors have been developed. A number of issues motivated these developments:

- Calculating the global minimum RMSD can be computationally difficult.
- The RMSD changes continuously but not smoothly as the coordinates of one structure are smoothly varied, because there are discontinuities in the gradient when the optimal permutation changes.
- The RMSD does not always accurately capture the degree of (dis)similarity between two structures in the most useful way.[10,14]
- The RMSD can only be used to compare structures with the same number of equivalent atoms.

We now briefly review some of the metrics and methods that are related to the approaches developed in the present work.

Gaussian Kernels A variety of algorithms (including those we will employ below) are based on the definition of a density function using a sum of Gaussian kernels of width σ_G,

$$\rho_p(\mathbf{r}) = \sum_{j=1}^{N} \exp\left(\frac{|\mathbf{r} - \mathbf{r}_j^p|^2}{2\sigma_G^2}\right), \qquad \rho_q(\mathbf{r}) = \sum_{j=1}^{N} \exp\left(\frac{|\mathbf{r} - \mathbf{r}_j^q|^2}{2\sigma_G^2}\right). \tag{6}$$

These densities are equivalent to the kernel functions used in the kernel correlation point set registration methods.[19,20,30] The properties of the overlap integral with respect to a set of rigid body motions, $T : \mathbb{R}^3 \to \mathbb{R}^3$,

$$\Omega^{pq}(\mathbf{T}) = \iiint \rho_p(\mathbf{r})\rho_q(T(\mathbf{r}))\mathrm{d}\mathbf{r}, \tag{7}$$

are considered. The Gaussian kernel is one of the more common kernels used to generate a density function from a list of coordinates, but others have been proposed.[41–43] This functional representation of the densities is permutationally invariant and smooth.

Smooth Overlap of Atomic Positions (SOAP) This descriptor compares the local environment of two atoms by centering the density functions on two specific atoms and then evaluating the overlap integral with respect to all possible rotations.[41] The calculation can be performed efficiently by expressing the densities as truncated sums of spherical harmonics, whose integrals can be evaluated analytically to obtain density functions. This procedure allows the local environment of different atoms to be compared. There are a variety of ways that the local similarity metrics can be combined to determine the global similarity of two structures.[14] This method has been used to improve potential energy surface fitting within the Gaussian approximation potentials (GAP) framework.[44,45]

Maximum Overlap of Kernels The global maximum value of eq. (7) with respect to rotations has been used as an alternative metric for clusters, through searches based on simulated annealing.[42]

Fingerprint functions Fingerprint functions produce vectors of structural properties that are invariant to symmetries of the Hamiltonian. These properties are often based on the eigenvalues of various matrices associated with the structure, such as Coulomb matrices,[46] or kernel overlap matrices.[13,47] The norm of the ordered eigenvalues can be used as a metric, and if the vector is larger than the number of degrees of freedom it can provide a unique identifier for the structure.[13] Properties of the interatomic distance matrix have also been used to construct descriptors and metrics.[48]

Other Metrics A variety of other metrics have been developed, based on a number of properties, including bond-order parameters,[44,49–51] "similarity functions",[52] bond network graphs,[53] localised Coulomb representations (related to Coulomb matrices in the same way as SOAP relates to kernel overlap matrices),[54] and radial distribution functions.[55] The similarity of proteins has been calculated by projecting the shape of the protein as an expansion of Wigner-D functions and calculating the corrlelation between these expansions.[56]

2 New Alignment Algorithms

In this paper we present two algorithms for aligning structures. A brief overview of both methods is given below; full descriptions can be found in appendices A and B.

2.1 Go-PERMDIST

This method extends the branch and bound Go-ICP[21] algorithm in contribution with PERMDIST, and we therefore refer to it as *Globally Optimised PERMDIST*. It achieves deterministic calculation of the global minimum RMSD in polynomial time, providing an alternative

to running PERMDIST from multiple random starting orientations. We find that the full run time of the algorithm for two structures increases rapidly with the RMSD between them.

The calculation of the lower bound, described in full in appendix A.2, uses a different approach from previous branch and bound methods,[21,29] based on the spherical law of cosines to bound the magnitude of the relative rotation within a given search region.

2.2 FASTOVERLAP

This method is a variant of kernel correlation[19] based alignment methods. It uses a fast Fourier transform or fast $SO(3)$ Fourier transform[31] to find the maximum correlation/overlap between density/kernel representations of both periodic structures and clusters deterministically. FASTOVERLAP is also related to the methods proposed by Bartók et al.[41], Ferré et al.[42] and Makadia et al.[20]. In appendices B.1 and B.3 we demonstrate how the kernel correlation/overlap can be used to estimate the global minimum RMSD efficiently for structures that are reasonably close for periodic systems and isolated clusters of atoms; for alignments with large RMSDs the inherent approximations will break down. The maximum correlation displacement/rotation can be used as a starting point for the PERMDIST algorithm. The run time for FASTOVERLAP is not very sensitive to the RMSD for a given system.

The FASTOVERLAP algorithm requires us to choose the width of the Gaussian kernels, σ_G. Our investigations have shown that setting σ_G equal to 1/3 of the interatomic separation gives generally good performance. For cluster alignment the angular momentum cutoff, $l_{\max}$, needs to be set as well, which defines the angular resolution, $\Delta_\theta = \pi/l_{\max}$ of the SOFT to find the global maximum kernel correlation. For most purposes setting $l_{\max} = 15$ worked well, though for large systems, the algorithm may display improved performance if the angular momentum cutoff is higher.

In appendix B.2.3 we show that the computational complexity of the periodic algorithm scales as $O(N^2)$ with the number of identical atoms. In appendix B.3 we show the complexity scales as $O(N^{5/3}l_{\max}^3 + l_{\max}^4 + N^2)$ for clusters.

In appendices B.3.1 and B.3.2 we also describe two computational methods for calculating the SO(3) Fourier coefficients more efficiently, which may be used to decrease the run time of the FASTOVERLAP algorithm or SOAP based similarity metrics.[41]

3 Performance of Alignment Algorithms

The performance of various alignment algorithms was assessed by comparing the lowest RMSD values and associated computational cost for a test set of structures. Timing benchmarks correspond to a single CPU core on a workstation with an Intel 3.3 GHz i7 Haswell processor. We primarily benchmarked the new algorithms against the methods in the Cambridge Energy Landscapes software package, as these methods perform significantly better than alternative algorithms, whose performance is discussed in in section 3.3.

3.1 Periodic Systems

Three different algorithms for aligning periodic systems in OPTIM,[36] corresponding to keywords PERMDIST, ATOMMATCHFULL and ATOMMATCHDIST, were tested against the FASTOVERLAP kernel correlation/Gaussian overlap schemes. The algorithms were tested on amorphous local minima for a binary Lennard-Jones liquid containing 204 atoms of type A and 52 atoms of type B, with a density of $1.2\,\sigma_{AA}^{-3}$. The energies were calculated using the usual Lennard-Jones pair potential with the Stoddard–Ford quadratic cutoff.[57] The interaction parameters used were $\epsilon_{AA} = 1.0$, $\epsilon_{AB} = 1.5$, $\epsilon_{BB} = 0.5$, $\sigma_{AA} = 1.0$, $\sigma_{AB} = 0.8$ and $\sigma_{BB} = 0.88$, corresponding to a popular model glass former.[58]

For the FASTOVERLAP algorithm we set the kernel width to 1/3 of the average interatomic spacing, $\sigma_G = \sigma_{AA}/(3\sqrt[3]{1.2}) = 0.314\,\sigma_{AA}$, and the cutoff wavevector order to 6.

Data Generation Two data sets of 100 unique minima were generated using the Python Energy Landscape Explorer (PELE)[59] and used to compare the algorithms. The first set was

created by a basin-hopping[60–62] global optimisation run from a random starting point. The second data set was generated by taking random steps away from one particular minimum, using a local geometry optimisation after each step to locate new minima. The steps were performed by assigning every atom a uniform random displacement along each axis of up to $0.3\,\sigma_{AA}$. The minimum RMSD for every pair of minima in each data set was calculated using several different alignment algorithms to compare them. Because these schemes are not necessarily symmetric in their arguments, all of the 10,000 possible pairs of minima were used.

The above procedure for generating the minima tends to produce pairs of structures that are already reasonably well aligned. A naive calculation of the RMSD without any form of alignment often produces an RMSD very close to the optimal value. Hence each minimum was also scrambled by applying a random global translation and permutation.

Performance on Scrambled Data A graphical comparison of the performance for the scrambled data sets is shown in fig. 1. The lower the RMSD, the better the alignment. The fourth column shows the best RMSD located, so if FASTOVERLAP always calculated the lowest RMSD it would give a straight line for that column. The percentage of RMSDs found by each method within a certain tolerance of the best RMSD is shown in fig. 2.

For RMSD $< 0.6\,\sigma_{AA}$ the FASTOVERLAP method always found an alignment quite close to the best RMSD. However, for a small number ($\sim 1\%$) of more distant pairs of minima it fails. For these structures the RMSD is dominated by atoms that are separated by $> 1\,\sigma_{AA}$, so the approximations made in the derivation are expected to fail and optimising the overlap no longer corresponds to optimising the RMSD. These results show that when aligning reasonably close minima, the FASTOVERLAP method is very reliable and is significantly better than the other methods.

All the other methods show significantly worse performance than FASTOVERLAP, except for the more distant pairs of minima, often failing to identify relatively close minima

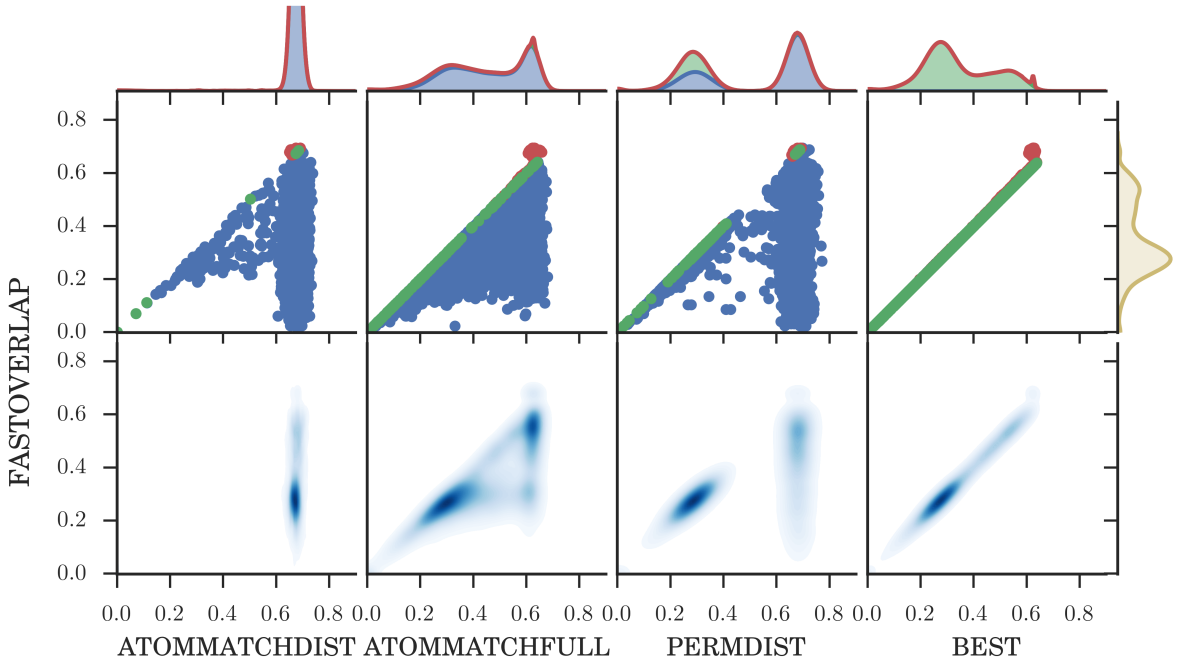

Figure 1: Comparison of the RMSD/σ_{AA} calculated by FASTOVERLAP against the RMSD found by various different alignment algorithms for scrambled amorphous binary Lennard-Jones structures. BEST gives the lowest RMSD found by any means. The top row shows a scatter plot of the RMSD found by FASTOVERLAP against the RMSD found by the methods listed on the bottom; red, green and blue points indicate whether the FASTOVERLAP method found a higher, equal or lower RMSD. The bottom row shows the density distribution of the scatter plots. Above the scatter plots the marginal distribution of the RMSD found by the methods listed below are shown. On the right next to the scatter graphs the marginal distribution of the RMSD found by FASTOVERLAP is shown. All the marginal distributions are on the same scale.

reliably. ATOMMATCHDIST does not identify the lowest RMSD for the vast majority of minima. PERMDIST is slightly better at identifying relatively close minima than ATOMMATCHFULL, but failed to find the global minimum for nearly all the pairs separated by intermediate distances, where ATOMMATCHFULL performs slightly better.

The bimodal distribution of RMSD found is due to the two different methods for generating minima. The dataset that produced minima by stepping from the same minimum repeatedly tended to generated very similar structures, while the basin-hopping run would tend to produce more diverse structures.

We also note that for this system the RMSD is peaked around $0.7\,\sigma_{AA}$, as this is around

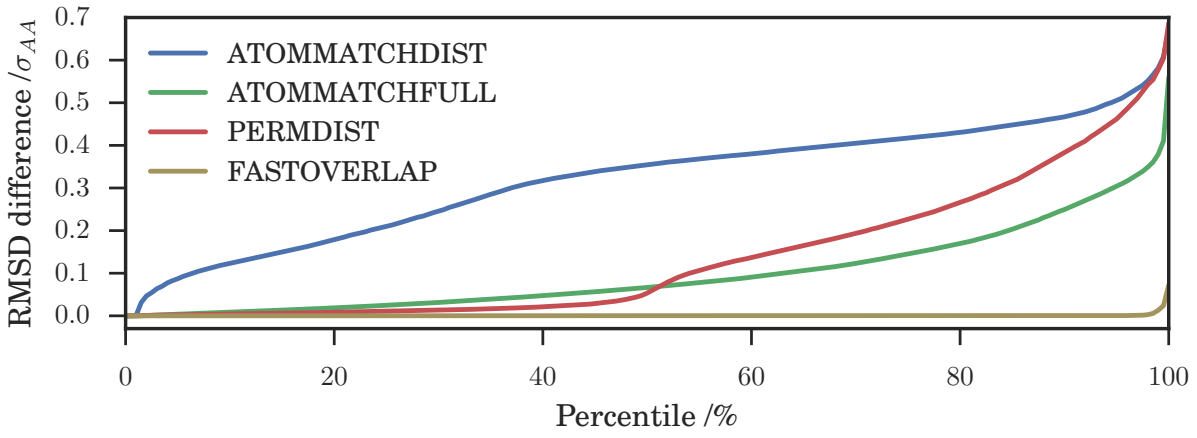

Figure 2: Graph comparing the accuracy of the different methods for aligning the scrambled binary Lennard-Jones structures, plotting the difference between the calculated RMSD and the optimal value against the percentage of alignments with a smaller difference.

the maximum the RMSD of the system can be after solving the assignment problem. For a bad alignment the atomic separations will be approximately evenly distributed between 0 and 1, resulting in an RMSD of around $0.7\,\sigma_{AA}$. When one of the alignment algorithms fails to find the correct translational alignment then it will return an RMSD of around $0.7\,\sigma_{AA}$, so worse methods will have larger peaks at $0.7\,\sigma_{AA}$ due to having more failed alignments.

Computational Complexity To measure the computational complexity the time to perform the alignment was calculated for supercells of increasing size. The potential used in this case corresponds to a single atomic species with pairwise Lennard-Jones interactions and fixed number density $1.05\,\sigma_{AA}$. System sizes ranging from 128 to 16384 atoms were tested. The average times taken for the different sized systems are shown in fig. 3; we observe an asymptotic approach to $O(N^2)$ scaling, as suggested by the analysis in appendix B.2.3.

Calculating all 10,000 alignments with the FASTOVERLAP algorithm required 90 s.

Go-PERMDIST We found that Go-PERMDIST requires a significantly larger runtime than FASTOVERLAP to achieve comparable performance. The run time for FASTOVERLAP was approximately equivalent to around five steps of the Go-PERMDIST algorithm,

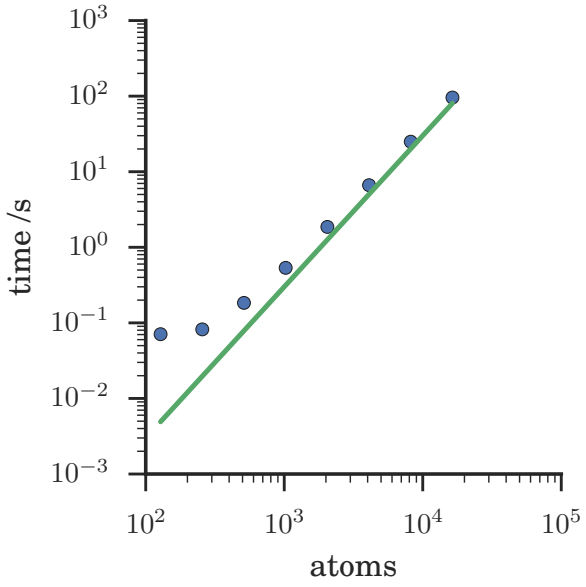

Figure 3: Average time required to calculate the RMSD using the FASTOVERLAP algorithm for periodic binary Lennard-Jones minima at a number density of 1.05. The green line shows an N^2 relationship.

which normally needed 100–1000 steps to find the optimal alignment.

3.2 Clusters

3.2.1 FASTOVERLAP

For aligning clusters the algorithm corresponding to the keyword PERMINVOPT in GMIN[35] was compared to the FASTOVERLAP algorithm for clusters. PERMINVOPT is the same algorithm as PERMDIST, but it also tests alignment for inverted structures. The maximum number of iterations in the PERMDIST algorithm was varied from 300 to 3000 to evaluate the effect of this parameter on the alignment. The algorithms were compared for Lennard-Jones (LJ) clusters of 38 atoms, LJ_{38}, using a database of 1000 unique minima generated in a discrete path sampling study.[6,7] Minimum RMSD values were calculated for all pairs.

For the FASTOVERLAP algorithm the kernel width was set to approximately 1/3 of the average interatomic spacing, $\sigma_G = 0.3\sigma$, where $2^{1/6}\sigma$ is the Lennard-Jones equilibrium

separation. The cutoff angular momentum degree was set to 15.

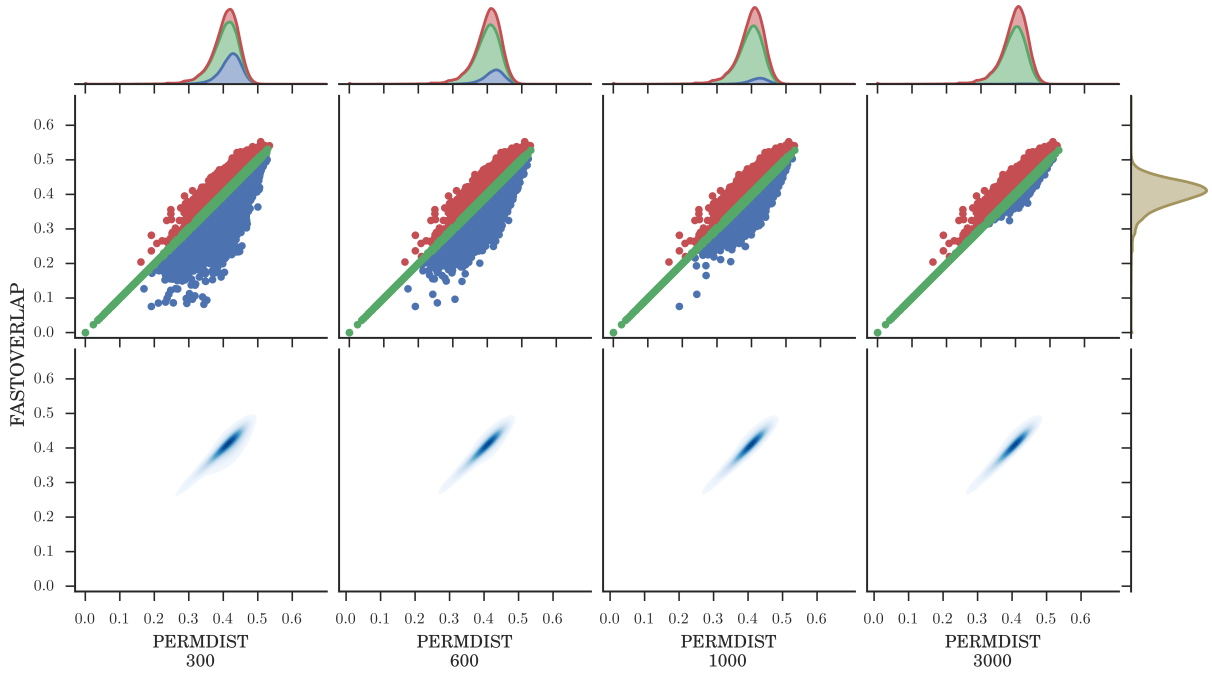

Figure 4: Comparison of the RMSD calculated by FASTOVERLAP against the RMSD found by PERMDIST for clusters of 38 Lennard-Jones atoms as a function of the number of PERMDIST iterations. The top row shows a scatter plot of the RMSD found by FASTOVERLAP against the RMSD found by PERMDIST on the bottom; red, green and blue points indicate whether the FASTOVERLAP method found a higher, equal or lower RMSD. The bottom row shows the density distribution of the scatter plots. Above the scatter plots the marginal distribution of the RMSD is illustrated. On the right, next to the scatter graphs, the marginal distribution of the RMSD found by FASTOVERLAP is shown. All the marginal distributions are on the same scale.

Performance A comparison of the performance of PERMDIST and FASTOVERLAP is shown in fig. 4. The percentage of RMSDs found by each method within a certain tolerance of the best RMSD is shown in fig. 5.

FASTOVERLAP finds the optimal RMSD for about 71% of the pairs of minima tested, and always finds the optimal RMSD for pairs separated by less than 0.15σ. After 600 iterations PERMDIST has nearly identical performance to FASTOVERLAP, with FASTOVERLAP performing slightly better for closer pairs of structures. After 1000 iterations PERMDIST is better, the same or worse than FASTOVERLAP for 26%, 67% or 7% of the

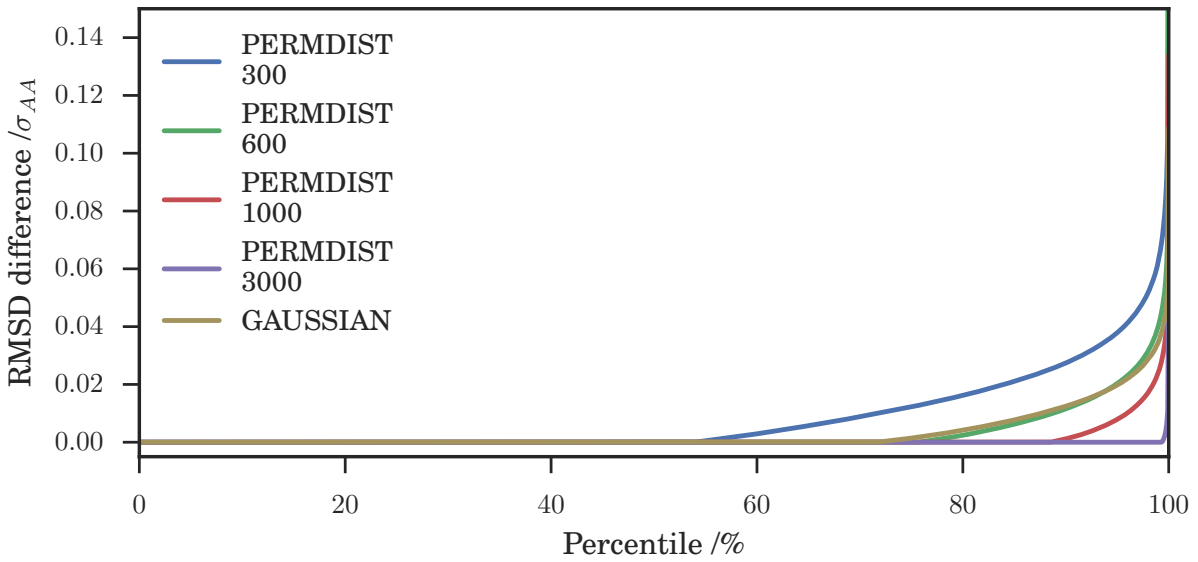

Figure 5: Graph comparing accuracy of the different methods for aligning LJ_{38} clusters, showing the percentage of alignments that achieved within a certain RMSD of the best found RMSD for each respective algorithm.

pairs of minima, and after 3000 iterations these figures change to 28%, 71% or 0.5% of the pairs.

The FASTOVERLAP algorithm failed to find the optimal RMSD for a few moderately close pairs of minima with 'non-cooperative' alignments,[63] where the difference in structure is dominated by a small number of atoms moving a relatively long distance. For these alignments choosing a larger kernel width generally resulted in finding the optimal alignment. For more distant minima FASTOVERLAP tended to fail because large numbers of atoms needed to be displaced a long way in the optimal alignment, so the assumptions made in the derivation do not hold (see appendix B.3).

Computational complexity A test set of random LJ minima was used to analyse the computational scaling of the algorithm with system size ranging from 128 to 8192 atoms. The results are for a kernel width of $\sigma_G = 0.3\sigma$ and angular momentum cutoff $l_{\max} = 15$. The timings of the calculations shown in fig. 6 confirm the expected $O(N^{5/3})$ scaling for fixed angular momentum cutoff, deduced in appendix B.3. The scaling with respect to

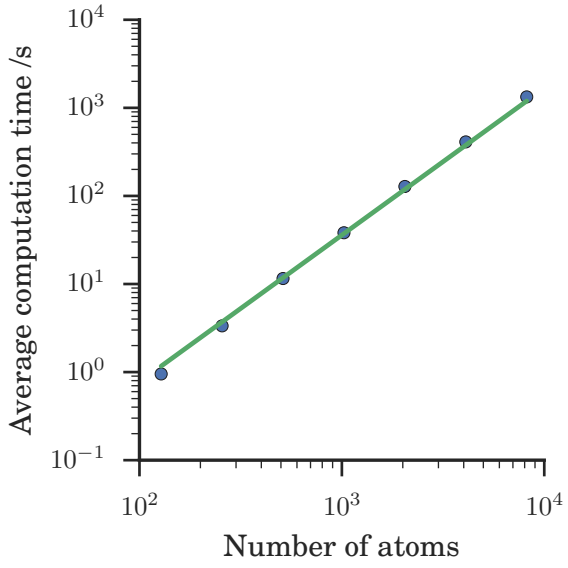

Figure 6: Average time required to align different sized structures using the FASTOVERLAP algorithm, with a fixed angular momentum cutoff of $l_{\max} = 15$, for random Lennard-Jones clusters. The green line shows an $N^{5/3}$ relationship.

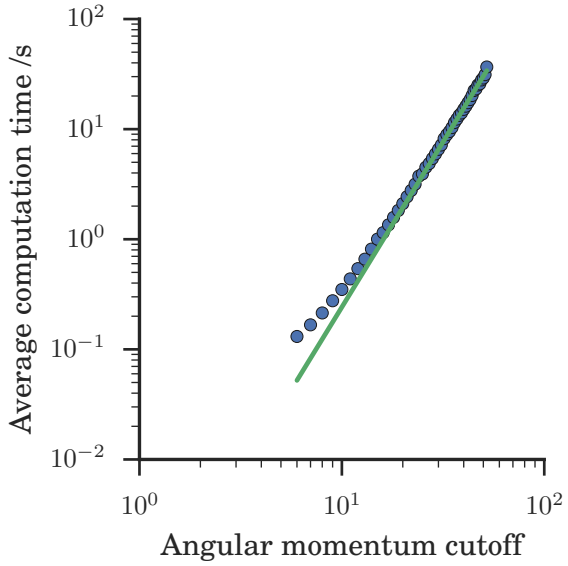

Figure 7: Average time required to align different sized structures using the FASTOVERLAP algorithm, for a range of angular momentum cutoffs, and random minima for Lennard-Jones clusters of 128 atoms. The green line shows an $l_{\max}^3$ relationship.

the angular momentum cutoff is shown in fig. 7; the $O(l_{\max}^3)$ behaviour suggests that the computational complexity is dominated by the $O(N^{5/3}l_{\max}^3)$ calculation of the SO(3) Fourier coefficients, rather than the $O(l_{\max}^4)$ cost of performing the inverse SO(3) Fourier transform (see appendix B.3).

3.2.2 Go-PERMDIST

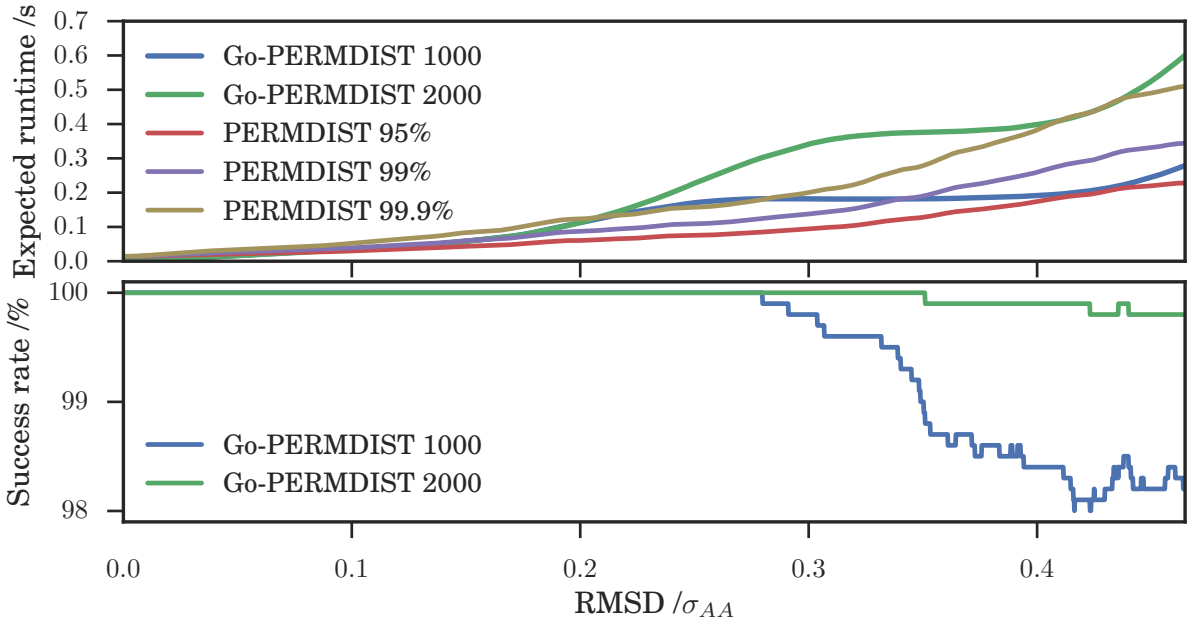

Figure 8: A comparison of the performance for Go-PERMDIST and PERMDIST over a range of alignments of LJ_{38} clusters. The top graph shows the expected runtime of Go-PERMDIST limited to 1000 or 2000 iterations, and the expected runtime of the PERMDIST algorithm to achieve a given success rate to find the global minimum RMSD. The bottom graph shows thea rolling-average success rate of the limited iteration Go-PERMDIST algorithm.

Although the Go-PERMDIST algorithm can calculate the minimal RMSD deterministically, for structures that are relatively distant this calculation can take an extremely long time. However if speed is critical, the Go-PERMDIST algorithm can be run with a maximum number of iterations to ensure that it terminates faster. This behaviour is offset by a slight loss in reliability.

For a pair of structures, if the random starting orientations are selected uniformly (which

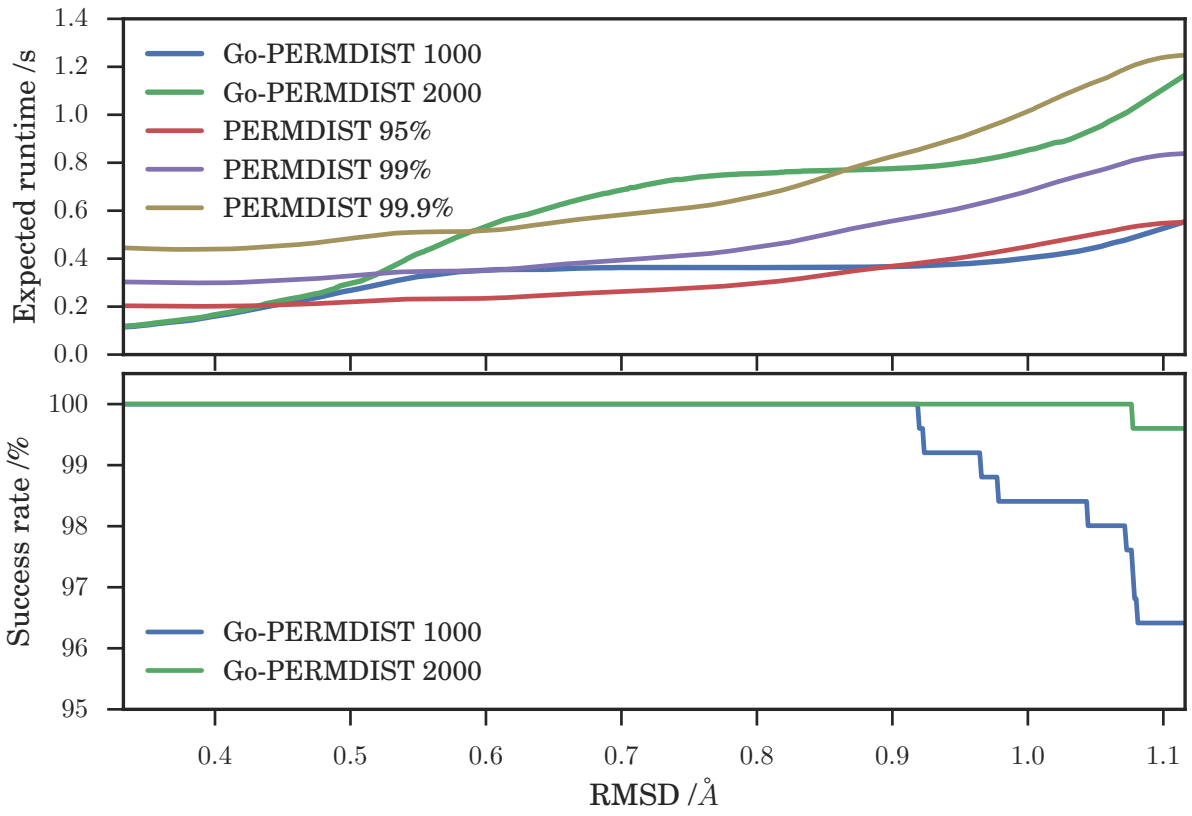

Figure 9: A comparison of the performance for Go-PERMDIST and PERMDIST over a range of alignments of Au_{55} clusters. The top graph shows the expected runtime of Go-PERMDIST limited to 1000 or 2000 iterations, and the expected runtime of the PERMDIST algorithm to achieve a given success rate to find the global minimum RMSD. The bottom graph shows thea rolling-average success rate of the limited iteration Go-PERMDIST algorithm.

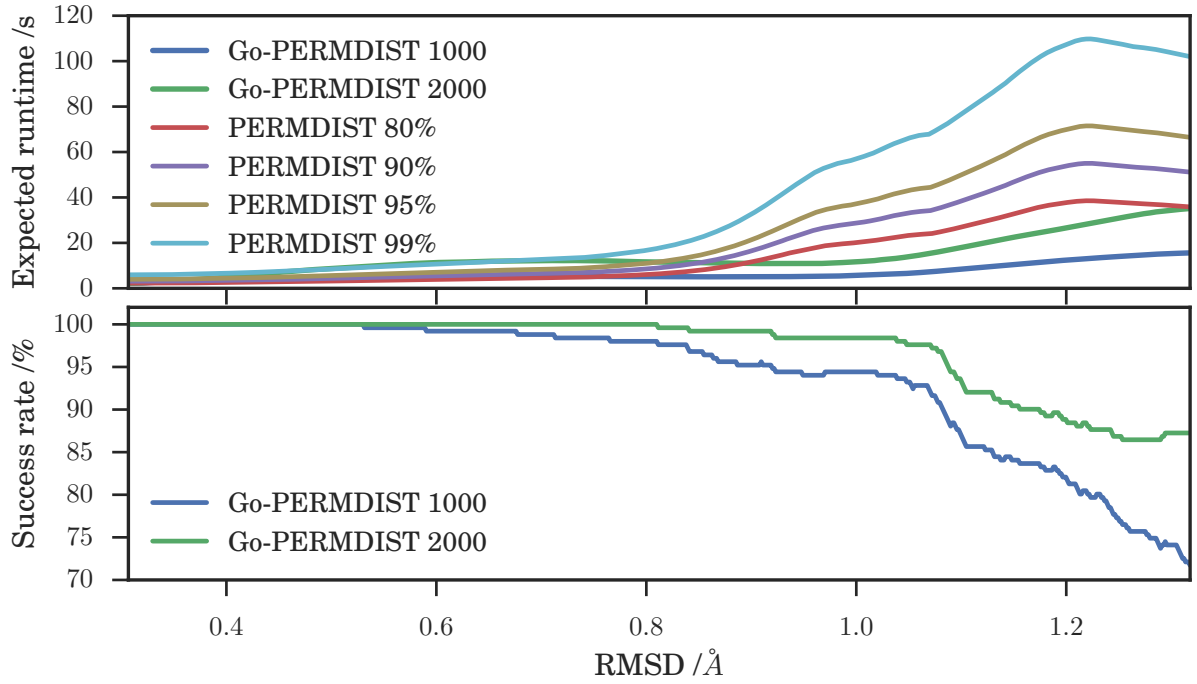

Figure 10: A comparison of the performance for Go-PERMDIST and PERMDIST over a range of alignments of Au_{147} clusters. The top graph shows the expected runtime of Go-PERMDIST limited to 1000 or 2000 iterations, and the expected runtime of the PERMDIST algorithm to achieve a given success rate to find the global minimum RMSD. The bottom graph shows thea rolling-average success rate of the limited iteration Go-PERMDIST algorithm.

is not an immediately straightforward task[64]), there will be a fixed probability, p_{find}, that PERMDIST will find the global minimum RMSD for a random starting orientation, which will be proportional to the size of the basin of attraction. The probability that the correct global minimum RMSD has been found after n random orientations will be,

$$p_{\text{found}}(n) = 1 - (1 - p_{\text{find}})^n, \tag{8}$$

and the number of random starting orientations to achieve a certain success rate, f_{success} will be,

$$n_{\text{rate}}(f_{\text{success}}) = \frac{\log(1 - f_{\text{success}})}{\log(1 - p_{\text{find}})}. \tag{9}$$

The maximum likelihood estimator of p_{find} is,

$$\hat{p}_{\text{find}} = \frac{N_{\text{trials}}}{\sum_{j=1}^{N_{\text{trials}}} n_j}, \tag{10}$$

for N_{trials} independent alignments, where PERMDIST finds the global minimum after n_j random starting orientations for alignment number j. For any given system size there is generally a fixed computation time for each iteration, so we can use eqs. (9) and (10), to estimate the time that PERMDIST would need to run for to achieve a given level of accuracy. It was found that p_{find} was strongly correlated with the distance between the structures.

Data Generation Three test sets were generated to compare the performance of PERMDIST and Go-PERMDIST.

LJ_{38} 6000 pairs of structures from the dataset used in section 3.2.1, with RMSD distributed from 0 to 0.6 σ.

Au_{55} 1000 pairs of structures from a dataset of the 100 lowest lying minima of 55 gold atoms, Au_{55}, modelled by the Gupta potential[65] and found by basin-hopping using GMIN,[35,60–62] with RMSD distributed from 0 to 1.3 Å.

Au_{147} 1000 pairs of structures from a dataset of the 100 lowest lying minima of 147 gold atoms, Au_{147}, modelled by the Gupta potential[65] and found by basin-hopping using GMIN,[35,60–62] with RMSD distributed from 0 to 1.5 Å.

Performance Graphs comparing the performance of Go-PERMDIST and PERMDIST are shown in figs. 8 to 10. Each figure shows a comparison for the estimated runtime between Go-PERMDIST limited to 1000 or 2000 iterations and the expected runtime of the PERMDIST algorithm to reach a certain level of accuracy for a given RMSD (estimated using eqs. (9) and (10) and the pairs with the most similar RMSD).

The two algorithms show comparable performance, and both find higher RMSD alignments more difficult. For PERMDIST, the number of random orientations that needed to be tested increased approximately as RMSD^2 for $\mathrm{RMSD} \gg 0$. The runtime for PERMDIST to achieve the same level of accuracy as Go-PERMDIST was generally higher or similar to the expected runtime of Go-PERMDIST.

Ensuring that rotations were sampled uniformly was important for many alignments with PERMDIST, especially for pairs of structures with distinct alignments but very similar RMSDs. For these structures the PERMDIST algorithm would often be attracted to the region around the alignment with a slightly higher RMSD, and so would take a disproportionately long time to find the best alignment.

3.3 Comparison to Permutation Optimisation Algorithms

We also tested the performance of our own implementations of other algorithms against the above methods, in particular the Monte Carlo permutation optimisation algorithm developed by Sadeghi et al.[13] and the branch and bound permutation optimisation algorithm developed by Hong et al.[34].

Both algorithms were found to scale exponentially with system size, which made the calculation of the minimal RMSD for systems with more than around 15 atoms much slower

than FASTOVERLAP, PERMDIST and Go-PERMDIST. This behaviour is expected as both algorithms optimise over an exponentially large space of permutations, whereas the FASTOVERLAP, PERMDIST and Go-PERMDIST algorithms are effectively 3D optimisation algorithms.

If the structures are initially relatively well aligned then the Monte Carlo permutation algorithm could occasionally find the optimal alignment relatively quickly. However, for systems with as few as 12 atoms it could take over 20,000 steps to find the optimal alignment, especially if the structures were not initially close. Our implementation of the algorithm was implemented in python using PELE.[59]

The branch and bound permutation optimisation algorithm was relatively efficient when aligning permutational isomers (or close to permutational isomers) compared to the other methods tested, because it is then easier to discard branches with the wrong permutation, so only a relatively small number of permutations need to be tested. However, the number of permutations required increases exponentially with RMSD, so the algorithm showed poor performance in general. Our implementation of this procedure was implemented in python, and to improve the performance we used the shortest-augmenting path algorithm to calculate the upper bounds of the branches.

4 Conclusions

We have shown that it is possible to estimate the RMSD between structures by calculating the maximum kernel correlation, so long as the interatomic separation is relatively large compared to the kernel size. We then demonstrated that the FASTOVERLAP algorithm can find the maximum value of the overlap in periodic and isolated systems efficiently and deterministically using fast Fourier transforms (FFT) and fast special orthogonal transforms (SOFT). Additionally, we have shown that it is possible to calculate the true RMSD of a system deterministically in a manner that scales polynomially with the number of atoms

and RMSD, using the branch and bound algorithm, Go-PERMDIST. The correct RMSD is often obtained when a early exit condition is applied.

For periodic systems FASTOVERLAP performs particularly well, and scales favourably with both system and database size for multiple alignment tasks. The algorithm reliably identifies the optimal alignment for pairs of structures that are reasonably close together. For more distant configurations the performance degrades as the assumptions underlying the derivation of the algorithm break down. However, for these structures it is likely that finding the minimum RMSD is less critical in applications. For periodic systems the Go-PERMDIST algorithm is significantly slower than the FASTOVERLAP algorithm.

For isolated clusters of atoms the FASTOVERLAP algorithm performs less well, while the PERMDIST and Go-PERMDIST procedures are relatively effective and efficient. The Go-PERMDIST algorithm shows comparable or better performance than PERMDIST with random restarts when using the early exit condition.

Future Work In certain applications we seek only the closest structures to a given target structure from a large database. In this situation it will be possible to adapt the Go-PERMDIST method to simultaneously align over the database and to quit once it has found the closest structures, instead of aligning the target structure with every member of the database.

When modelling the growth of clusters or mutations of proteins, it can be useful to align structures with different numbers of atoms. The Go-PERMDIST algorithm could be modified to perform this alignment, but the translational component would also have to be considered, in addition to the rotational alignment, as superimposing the centroids of the two structures no longer results in the optimal alignment. This step could be achieved by including translation alignment as in the Go-ICP method.[21,22]

It may also be possible to improve the performance of the PERMDIST algorithm using basin-hopping global optimisation[60–62] and taking smaller, non-random, rotational steps.

Care would be required to ensure that the procedure does not get stuck in a local minimum.

When aligning very large databases of structures it can be prohibitively expensive to align every possible pair. The FASTOVERLAP method allows the RMSD between structures to be estimated quickly as an alternative metric. It may be possible in the future to develop diagnostic statistics that could be used to give an indication whether FASTOVERLAP has found the optimal alignment or determine whether there is a better kernel width for a particular pair of structures.

The maximal kernel overlap found by FASTOVERLAP may also be useful when only the similarity between two structures is needed, for example when comparing configurations with different numbers of atoms. It also may be possible to generalise the calculation of the SO(3) Fourier coefficients to allow optimisation over translations in addition to rotations.

Recommended Usage If run time is not critical we recommend using the Go-PERMDIST algorithm, which is guaranteed to find the best RMSD for both periodic systems and clusters given enough time. If the run time is important, then the FASTOVERLAP algorithm should be used for periodic systems and the Go-PERMDIST algorithm with an early exit condition could be used for clusters. For biomolecules it is likely that other methods, for example LPERMDIST,[10] that take into consideration the local structure of the molecule, will be more effective.

Acknowledgement

We are grateful to Dr Dmitri Schebarchov for his help in providing the gold cluster datasets. This work was supported by the EPSRC Cambridge NanoDTC, EP/G037221/1. The program used to perform the alignments may be found in the Cambridge Energy Landscape Database, `http://www-wales.ch.cam.ac.uk/CCD.html`. Standalone Python and Fortran implementations of the alignment routines can be found on Github, `https://github.com/matthewghgriffiths/fastoverlap`. The graphs were made using Matplotlib (`https:`

//matplotlib.org/)[66] and Seaborn (http://seaborn.pydata.org/). The diagrams were produced using Inkscape (https://inkscape.org/en/).

A Go-PERMDIST

Here we describe the branch and bound algorithm developed to calculate the RMSD between two structures deterministically.

A.1 Deterministic Calculation of RMSD

It is possible to adapt the branch and bound algorithm developed by Yang et al.[21], Li and Hartley[29] to find the global minimum RMSD in polynomial time. To apply a branch and bound algorithm we need to parameterise the domain over which we are searching for a solution and define bounding functions that allow us to prune the search space of the algorithm. For isolated clusters we follow Li and Hartley[29] and use the angle-axis representation of rotations, where all possible rotations can be described as a point within a $\mathbb{R}^3$ sphere of radius π, and search within the minimal $[-\pi, \pi]^3$ bounding cube enclosing this sphere. Search regions that are totally outside this sphere can be discarded to reduce the search space by around a factor of two. For periodic systems the search space exactly corresponds to displacements within the crystal unit cell.

When searching for permutation-inversion isomers we can define a second search domain of the same size, corresponding to the set of rotations on the inverted structure. For periodic systems our search space can be defined simply within the unit cell. If we wish to find the global RMSD over all the point group symmetries of the supercell, then we can add extra search domains corresponding to translations within the unit cell for each point group operation. Additionally, if we are interested in finding the closest structure from a set of configurations to a target structure, we could treat each one as a separate search domain and search over all of them.

This version of branch and bound algorithm recursively explores the solution domain by breaking each cubic search region into eight smaller cubes, calculating the lower and upper bounds for the RMSD within each cube. If a cube is found to have a lower bound higher than the best found RMSD then the algorithm stops searching in that region of the domain, progressively eliminating areas of the search space. The process terminates once it finds a region where the upper bound is within a given tolerance of the lowest lower bound. The performance depends on our ability to accurately find lower and upper bounds for the search region. We now define functions that can be used to bound the RMSD for clusters and periodic systems.

A.2 Bounding RMSD for Clusters

For a given rotation matrix, $\mathbf{m}$, with corresponding angle-axis rotation, $\mathbf{v}$ and search region box width θ_{B}, the upper bound of the RMSD can be found by solving the permutational assignment problem between the target structure and rotated structure. Finding the lower bound of the RMSD requires finding the lower bound of the distance between all points in the structure within the search region. We can find a lower bound using the law of cosines, where points $\mathbf{r}_j^p$ and $\mathbf{m} \cdot \mathbf{r}_{j'}^q$ are separated by distance $d_{jj'}$ where,

$$d_{jj'}^2 = r_j^{p\,2} + r_{j'}^{q\,2} - 2r_j^p r_{j'}^q \cos\phi_{j,j'} \tag{11}$$

and $\phi_{j,j'}$ is the angle between $\mathbf{r}_j^p$ and $\mathbf{m}\cdot\mathbf{r}_{j'}^q$. We can calculate the lower bound of the distance between the points, $\underline{d}_{jj'}$ within the search region as,

$$\underline{d}_{jj'}^2 = \min_{|\theta|\leq\theta_{\mathrm{B}'}} r_j^{p\,2} + r_{j'}^{q\,2} - 2r_j^p r_{j'}^q \cos\left(\phi_{j,j'} + \theta\right). \tag{12}$$

where $\theta_{B'}$ is the maximum angle by which points can be rotated relative to the rotation $\mathbf{m}$ in the search box (see appendix A.2.1). Calculating,

$$\begin{aligned}\overline{\cos}(\phi_{j,j'}) &= \max_{|\theta|\leq\theta_{B'}} \cos\left(\phi_{j,j'}+\theta\right) \\ &= \begin{cases} 1, & \text{when } |\phi_{j,j'}| \leq \theta_{B'} \\ \cos\left(\phi_{j,j'}\right)\cos\left(\theta_{B'}\right), +|\sin\left(\phi_{j,j'}\right)|\sin\left(\theta_{B'}\right), & \text{when } |\phi_{j,j'}| > \theta_{B'}. \end{cases}\end{aligned} \tag{13}$$

This expression for $\overline{\cos}(\phi_{j,j'})$ does not require us to calculate the value of $\phi_{j,j'}$ because $\cos\left(\phi_{j,j'}\right)$ can be obtained from eq. (11) and $|\sin\left(\phi_{j,j'}\right)| = \sqrt{1-\cos\left(\phi_{j,j'}\right)^2}$. We can now calculate a lower bound for the distance between the two points,

$$\underline{d}^2_{jj'} = r_j^{p\,2} + r_{j'}^{q\,2} - 2r_j^p r_{j'}^q \overline{\cos}(\phi_{j,j'}). \tag{14}$$

This pairwise lower bound between all the points in both structures can be used by an assignment problem or nearest neighbour search algorithm to produce a lower bound for the RMSD in the bounding cube.

A.2.1 Composing Angle-Axis Rotations

To bound the pairwise distance we need to place a bound on the maximum angle by which a point could be displaced within the search box. The approach presented here differs from that used by Li and Hartley[29] and Yang et al.[21] They bound the maximum angle by which a point can differ after two rotations using the inequality that the angular distance between two rotations is less than or equal to the Euclidean distance between the vectors in the angle-axis representation. Here we consider how angle-axis rotations are composed to bound the maximum angle.

For an angle-axis rotation $\mathbf{v}+\mathbf{e}$ we want to find the magnitude of the rotation vector $\mathbf{e}'$ such that rotation by vector $\mathbf{v}$ then $\mathbf{e}'$ is equivalent to rotation by vector $\mathbf{v}+\mathbf{e}$. By

considering angle-axis rotations as arcs of a great circle their composition can be viewed as equivalent to vector addition of these arcs on the surface of a unit sphere (see fig. 11). The angle-axis rotation vector $\mathbf{v}$ has equivalent great circle arc AB, vector $\mathbf{v}+\mathbf{e}$ corresponds to arc AG, vector $\mathbf{e}'$ is equivalent to arc BC, and the composition of angle-axis rotations $\mathbf{v}$ followed by $\mathbf{e}'$ is $\mathbf{v}+\mathbf{e}$. The starting points of the arcs are arbitrary, so we have chosen A to correspond to the intersection of the great circles of the rotations $\mathbf{v}$ and $\mathbf{v}+\mathbf{e}$.

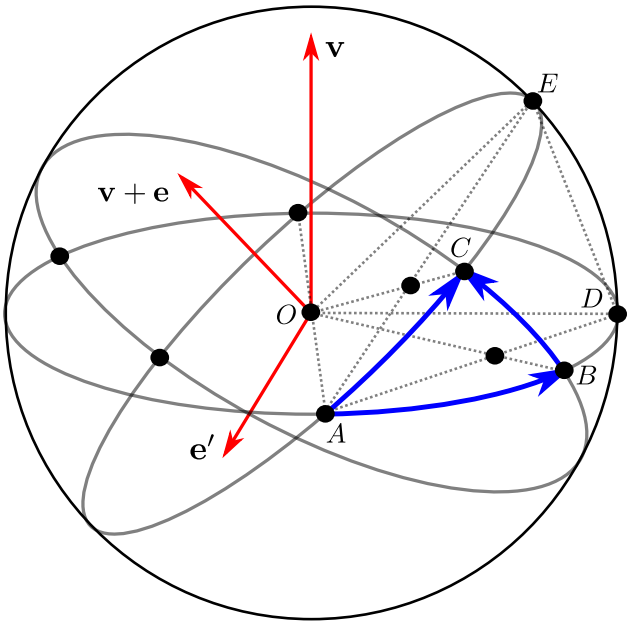

Figure 11: A diagram indicating how angle-axis rotations can be composed. The rotation corresponding to the angle axis vector $\mathbf{v}+\mathbf{e}$ is equivalent to the composition of the rotation around $\mathbf{v}$ then $\mathbf{e}'$. Alternatively, considering rotations as arcs of great circles, the arc AC is equal to the arc BC added to the arc AB.

To bound the distance between two points for the set of all possible rotations in a given search box, we find a bound on the arc $BC = |\mathbf{e}'|$ for the same search box, using the spherical law of cosines,

$$\cos BC = \cos AB \cos AC + \cos \angle DOE \sin AB \sin AC. \tag{15}$$

$\angle DOE$ is the angle between $\mathbf{v}+\mathbf{e}$ and $\mathbf{v}$, $AB = |\mathbf{v}|$ and $AC = |\mathbf{v}+\mathbf{e}|$. So we can bound

these values, for a search box centred on vector $\mathbf{v}$ with rotation $\theta_1 = |\mathbf{v}|$ and box width θ_B:

$$\cos \angle DOE \geq \frac{\theta_1}{\sqrt{\theta_1^2 + 3\theta_B^2/4}} = \cos \bar{\theta}_2, \tag{16}$$

$$AB = \theta_1 = |\mathbf{v}|, \tag{17}$$

$$\theta_1 - \frac{\theta_\mathrm{B}}{2} \leq AC \leq \theta_1 + \frac{\theta_\mathrm{B}}{2}. \tag{18}$$

From this result we can find a bound for the maximum angle, $\theta_{\mathrm{B}'}$, a point can be rotated within the search box,

$$\begin{aligned} \cos \theta_{\mathrm{B}'} &= \min[\cos AB] \\ &= \min \left[\cos \frac{\theta_B}{2}, \left(\cos(\theta_1)^2 + \cos \bar{\theta}_2 \sin(\theta_1)^2 \right) \cos \frac{\theta_B}{2} - (1 - \cos \bar{\theta}_2) \left| \sin \theta_1 \cos \theta_1 \sin \frac{\theta_B}{2} \right| \right]. \end{aligned} \tag{19}$$

This result can then be used in eq. (13) to obtain a lower bound for the distance between two coordinates inside the search box.

A.3 Bounding RMSD for Periodic Systems

For periodic systems we can follow a similar procedure, where for a given displacement, $\mathbf{d}$, with bounding box width d_B, the upper bound of the RMSD can be found by solving the assignment problem between the target structure and translated structure. To find the lower bound of the RMSD we need the lower bound of the distance between all the points in the structure, so if we employ the notation in appendix A.2, we can define the distance between points as,

$$d_{jj'} = \min_{\mathbf{l} \in \mathbf{L}} |\mathbf{r}_j^p - \mathbf{r}_{j'}^q - \mathbf{d} + \mathbf{l}| = |\mathbf{r}_j^p - \mathbf{r}_{j'}^q - \mathbf{d} + \mathbf{l}_{jj'}|, \tag{20}$$

where $\mathbf{l}_{jj'}$ is the lattice vector that minimises the distance between the points. The lower bound of the distance between the points is

$$\underline{d}_{jj'} = \begin{cases} 0, & d_{jj'} \leq \sqrt{3} d_{\mathrm{B}}/2, \\ d_{jj'} - \sqrt{3} d_{\mathrm{B}}/2, & d_{jj'} > \sqrt{3} d_{\mathrm{B}}/2. \end{cases} \tag{21}$$

This result can be used to calculate a lower bound for the RMSD using the assignment algorithm.

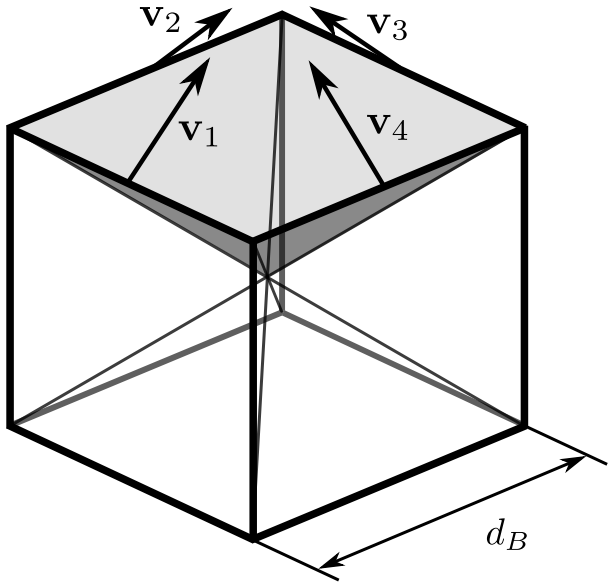

Figure 12: A diagram showing how the search cube can be split into six identical pyramids. The faces of the top pyramid have been shaded.

We can improve this lower bound by splitting the cube into six identical square pyramids. Consider one of these pyramids (for example the top one in fig. 12), with triangular face normals, $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, and $\mathbf{v}_4$. The closest distance of pairs of particles for which $F_i(\mathbf{r}_j^p, \mathbf{r}_{j'}^q) = (\mathbf{r}_j^p - \mathbf{r}_{j'}^q - \mathbf{d} + \mathbf{l}) \cdot \mathbf{v}_i \geq 0$, for $i = 1, 2, 3, 4$, will only ever be $d_{jj'}$, so we can define a new pair

wise distance lower bound for the pyramid as,

$$\underline{d}_{jj'} = \begin{cases} d_{jj'}, & \text{if } F_i(\mathbf{r}_j^p, \mathbf{r}_{j'}^q) > 0 \text{ for } i = 1,2,3,4, \\ & \text{else:} \\ 0, & d_{jj'} \leq \sqrt{3} d_{\mathrm{B}}/2, \\ d_{jj'} - \sqrt{3} d_{\mathrm{B}}/2, & d_{jj'} > \sqrt{3} d_{\mathrm{B}}/2. \end{cases} . \tag{22}$$

The lower bound for the box then can be found by calculating the lower bound for each pyramid and taking the minimum value.

A.4 Approximating Bounds

It requires $O(N^2)$ operations to solve the assignment problem, whereas using a k-dimensional binary search tree it is possible to find the set of nearest neighbours between two point clouds in $O(N \log N)$ operations.[67] We are only interested in the exact value of the upper bound if it is lower than any other upper bound found, so instead of solving the assignment problem for each search region we can perform an initial nearest neighbour search to give a lower bound for the upper bound search region, and then do the same calculation as in eqs. (12) and (14) to produce a lower bound for the RMSD. If the calculation of the nearest neighbour distance is found to give an upper bound less than the lowest found upper bound, then the more expensive calculation of the 'proper' upper bound using the assignment problem can be performed. Both the bounds calculated using the nearest neighbours approach will be at most equal to the bounds calculated using the assignment method.

A.5 Branch and Bound Algorithm

The branch and bound algorithm to find the global minimum RMSD, within a certain absolute tolerance, ϵ_a and relative tolerance, ϵ_r, uses a *best-first-search*, where the regions in search space with the smallest lower bounds are explored first. To describe the algorithm we

Algorithm 1 Go-PERMDIST

Input: $\mathbf{R}^p$, $\mathbf{R}^q$, $\mathcal{C}_G$, (*optional* $\mathcal{C}_G^*$) ▷ Structures to align and search region(s)
Output: $\overline{E}, \mathbf{v}_{\text{best}}$ ▷ RMSD and transformation vector
add $\{\mathcal{C}_G^*\}$ to $\mathcal{Q}$
if Testing for Symmetries **then**
  add $\mathcal{C}_G^*$ to $\mathcal{Q}$
end if
$\overline{E} = +\infty$ ▷ Estimate of RMSD of p and q
loop
  get search cube $\mathcal{C}_t$ with lowest lower bound $\underline{E}(t)$ from $\mathcal{Q}$
  if $\overline{E} - \underline{E}(t) < \epsilon_a + \epsilon_r \underline{E}(t)$ **then**
    quit loop ▷ Stop algorithm once desired precision achieved.
  end if
  for 8 sub-cubes $\mathcal{C}_{t^*}$ of $\mathcal{C}_t$ **do**
    compute $\lfloor \overline{E} \rfloor(t)$
    if $\lfloor \overline{E} \rfloor(t) < \overline{E}$ **then** ▷ If estimate of upper bound less than current best upper bound, calculate upper bound using assignment algorithm.
      compute $\underline{E}(t)$
      if $\underline{E}(t^*) < \overline{E}$ **then**
        compute $\overline{E}(t^*)$
        if $\overline{E}(t^*) < (1+\epsilon_r)\overline{E}$ **then**
          use PERMDIST algorithm to refine $\overline{E}(t^*)$ and $\mathbf{v}_{t^*}$ to $\overline{E}^*$ and $\mathbf{v}^*$
          $\overline{E}$, $\mathbf{v}_{\text{best}} = \overline{E}^*$, $\mathbf{v}^*$ ▷ Update best estimate of RMSD
        end if
        add $\mathcal{C}_{t^*}$ to $\mathcal{Q}$
      end if
    else
      compute $\lfloor \underline{E} \rfloor(t)$
      if $\lfloor \underline{E} \rfloor(t) < \overline{E}$ **then**
        add $\mathcal{C}_{t^*}$ to $\mathcal{Q}$
      end if
    end if
  end for
end loop

first define some terms. We seek the globally optimum transformation vector $\mathbf{v}_G$ contained within a cubic search domain $\mathcal{C}_G$ of width w_G. For a periodic system $\mathbf{v}_G$ corresponds to the displacement vector and $w_G = L$, while for a cluster $\mathbf{v}_G$ corresponds to the rotation vector, $w_G = 2\pi$, and $\{\mathcal{C}_G^*\}$ are the set of search regions corresponding to point group symmetries of the system (e.g. inversion).

For a search domain, $\mathcal{C}_t$, centred at $\mathbf{v}_t$ and width w_t, we define the upper and lower bound of the RMSD to be $\overline{E}(t)$ and $\underline{E}(t)$. We define the upper and lower bound calculated by the nearest-neighbours approach as $\lfloor\overline{E}\rfloor(t)$ and $\lfloor\underline{E}\rfloor(t)$. We store the set of search regions in a priority queue, $\mathcal{Q}$, where the search region with the lowest lower bound, $\underline{E}(t_{\text{low}})$, is returned first. A detailed description is given in algorithm 1.

A.5.1 Asymptotic Behaviour

As the size of the search regions decreases the difference between the upper and lower bounds also decreases. For clusters we can see that with regions of angular size, θ_B, where $\theta_B \ll 1$,

$$d_{j,j'}^2 - \overline{d}_{j,j'}^2 \propto \theta_B r_j^p r_{j'}^q. \tag{23}$$

So the difference between the lower bound and upper bound will be proportional to θ_B. For periodic systems with regions of size, d_B, the difference between the upper and lower bound will be proportional d_B when $d_B \ll L/N^{1/3}$. The width of the search region is therefore proportional to the uncertainty in the lower bound. This result holds both when calculating the bounds using the assignment problem or the nearest-neighbours algorithm, so as the width of the search region decreases the difference between the bounds will decrease uniformly. This decrease guarantees that the global RMSD is found because the lowest upper bound calculated will always correspond to a possible RMSD alignment between the structures.

B FASTOVERLAP methods

Here we describe the mathematics underlying the FASTOVERLAP method. In appendix B.1 we show how the RMSD between two structures can be estimated by evaluating the kernel correlation between two structures and in appendix B.2 we show how we can find the global maximum of the kernel correlation efficiently by using the fast Fourier transform. We extend this method to apply to clusters in appendix B.3 using the discrete SO(3) Fourier transform (SOFT).

B.1 RMSD Estimation by Gaussian Overlap

Under certain circumstances it is possible to estimate the RMSD between two closely aligned structures by calculating the overlap integral of a set of Gaussian functions centred on the atomic coordinates. Consider a pair of atoms, specified by two Gaussian kernels with width σ_G, centred at positions $\mathbf{r}_0$ and $\mathbf{r}_1$,

$$\rho_0(\mathbf{r}) = \exp\left(-\frac{|\mathbf{r}-\mathbf{r}_0|^2}{2\sigma_G^2}\right), \qquad \rho_1(\mathbf{r}) = \exp\left(-\frac{|\mathbf{r}-\mathbf{r}_1|^2}{2\sigma_G^2}\right). \tag{24}$$

The overlap integral of these two Gaussians is,

$$\begin{aligned}\iiint \rho_0(\mathbf{r})\rho_1(\mathbf{r})\mathrm{d}\mathbf{r} &= \iiint \exp\left(-\frac{|\mathbf{r}_0-\mathbf{r}_1|^2}{4\sigma_G^2}\right)\exp\left(-\frac{|\mathbf{r}-\frac{\mathbf{r}_0+\mathbf{r}_1}{2}|^2}{\sigma_G^2}\right)\mathrm{d}\mathbf{r} \\ &= \exp\left(-\frac{|\mathbf{r}_0-\mathbf{r}_1|^2}{4\sigma_G^2}\right)(\pi\sigma_G^2)^{3/2}.\end{aligned} \tag{25}$$

When $|\mathbf{r}_0-\mathbf{r}_1| \ll \sigma_G$ we can approximate eq. (25) as,

$$\iiint \rho_0(\mathbf{r})\rho_1(\mathbf{r})\mathrm{d}\mathbf{r} \approx (\pi\sigma_G^2)^{3/2} - (\pi\sigma_G^2)^{3/2}\frac{|\mathbf{r}_0-\mathbf{r}_1|^2}{4\sigma_G^2} + o\left(\left(\frac{|\mathbf{r}_0-\mathbf{r}_1|}{\sigma_G}\right)^4\right). \tag{26}$$

Hence the overlap integral is proportional to the squared displacement between the atoms when they are close relative to σ_G. We can extend this result to estimate the RMSD of two

closely aligned periodic structures, p and q, by defining the density functions

$$\rho_p(\mathbf{r}) = \sum_{\mathbf{l}\in\mathbf{L}}\sum_{j=1}^{N} \exp\left(\frac{|\mathbf{r}-\mathbf{r}_j^p+\mathbf{l}|^2}{2\sigma_G^2}\right), \qquad \rho_q(\mathbf{r},\mathbf{d}) = \sum_{\mathbf{l}\in\mathbf{L}}\sum_{j=1}^{N} \exp\left(\frac{|\mathbf{r}-\mathbf{r}_j^q-\mathbf{l}-\mathbf{d}|^2}{2\sigma_G^2}\right). \quad (27)$$

Recall that $\mathbf{d}$ is the global displacement vector and $\mathbf{l}$ is a particular lattice vector. Using eq. (25) we can calculate the overlap integral or *kernel correlation* of these densities

$$\Omega^{pq}(\mathbf{d}) = \iiint_0^L \rho_p(\mathbf{r})\rho_q(\mathbf{r},\mathbf{d})\mathrm{d}\mathbf{r} = (\pi\sigma_G^2)^{3/2}\sum_{\mathbf{l}\in\mathbf{L}}\sum_{j=1}^{N}\sum_{j'=1}^{N} \exp\left(-\frac{|\mathbf{r}_j^p-\mathbf{r}_{j'}^q-\mathbf{l}-\mathbf{d}|^2}{4\sigma_G^2}\right). \quad (28)$$

We note that eq. (28) is invariant to permutations. If we then assume that $\sigma_G \ll r_{\mathrm{sep}}$, where r_{sep} is the minimum atomic separation, then for $\mathbf{d}$, $\mathbf{P}$ and $\mathbf{L}$ that minimise the RMSD we can approximate the integral above as

$$\Omega^{pq}(\mathbf{d}) \approx (\pi\sigma_G^2)^{3/2}\sum_{j=1}^{N} \exp\left(-\frac{|\mathbf{r}_j^p-\sum_{j'=1}^{N}\mathbf{P}_{jj'}(\mathbf{r}_{j'}^q-\mathbf{l}_{j'}-\mathbf{d})|^2}{4\sigma_G^2}\right). \quad (29)$$

If we also assume that $|\mathbf{r}_j^p-\sum_{j'=1}^{N}\mathbf{P}_{jj'}(\mathbf{r}_{j'}^q-\mathbf{l}_{j'}-\mathbf{d})| \ll \sigma_G$ for all j, so the structures are relatively similar, then if $\mathbf{d}_{\mathrm{m}} = \arg\max \Omega^{pq}(\mathbf{d})$,

$$\begin{aligned}\Omega^{pq}(\mathbf{d}_{\mathrm{m}}) &\approx (\pi\sigma_G^2)^{3/2}\sum_{j=1}^{N}\left(1-\frac{|\mathbf{r}_j^p-\sum_{j'=1}^{N}\mathbf{P}_{jj'}(\mathbf{r}_{j'}^q-\mathbf{l}_{j'}-\mathbf{d}_{\mathrm{m}})|^2}{4\sigma_G^2}\right)\\ &= N(\pi\sigma_G^2)^{3/2} - \frac{\pi^{3/2}\sigma_G^{-1/2}\sqrt{N}}{4}\,\mathrm{RMSD}(p,q). \end{aligned} \quad (30)$$

Hence the global maximum of $\Omega^{pq}(\mathbf{d})$ corresponds to the displacement that gives the minimum RMSD if the structures can be aligned sufficiently closely. Once the displacement is known, the corresponding optimal permutation matrix can be calculated using the Hungarian algorithm[24] or shortest augmenting path algorithm.[12]

The choice of σ_G is important for determining the accuracy of this method, If σ_G is set too small then the approximations only hold true for very closely aligned systems, while if σ_G

is too large then the value of the integral is no longer determined by the nearest neighbours at optimal alignment. In practice we found that setting σ_G to be 1/3 of the equilibrium pair separation produced good results over the widest range of structures.

B.2 Finding the Global Maximum of the Overlap Integral

To identify the global maximum of $\Omega^{pq}(\mathbf{d})$ efficiently we use Parseval's theorem, which states that for functions with Fourier series

$$\rho(\mathbf{r})_p = \sum_{\mathbf{k}\in\mathbf{K}} c^p_{\mathbf{k}} \exp\left(i\mathbf{k}\cdot\mathbf{r}\right), \qquad \rho(\mathbf{r},\mathbf{d})_q = \sum_{\mathbf{k}\in\mathbf{K}} c^q_{\mathbf{k}}(\mathbf{d}) \exp\left(i\mathbf{k}\cdot\mathbf{r}\right), \tag{31}$$

where $\mathbf{K}$ is the set of allowed wavevectors, so if $\mathbf{k} \in \mathbf{K}$, then $\mathbf{k} = 2\pi\mathbf{n}/L$, where $\mathbf{n} \in \mathbb{Z}^3$. Hence

$$\Omega^{pq}(\mathbf{d}) = \iiint_0^L \rho_p(\mathbf{r})\rho_q(\mathbf{r},\mathbf{d})\mathrm{d}\mathbf{r} = \frac{1}{L^3}\sum_{\mathbf{k}\in\mathbf{K}} c^p_{\mathbf{k}} c^q_{\mathbf{k}}(\mathbf{d})^*, \tag{32}$$

where $*$ indicates complex conjugation. The Fourier series coefficients can easily be calculated by treating the structure as a sum of delta functions at the atomic coordinates convolved with a Gaussian function with width σ_G

$$c^p_{\mathbf{k}} = e^{-\frac{1}{2}|\mathbf{k}|^2\sigma_G^2}\sum_{j=1}^{N} e^{-i\mathbf{k}\cdot\mathbf{r}^p_j} = e^{-\frac{1}{2}|\mathbf{k}|^2\sigma_G^2} d^p_{\mathbf{k}}, \tag{33}$$

$$c^q_{\mathbf{k}}(\mathbf{d}) = e^{-\frac{1}{2}|\mathbf{k}|^2\sigma_G^2} e^{-i\mathbf{k}\cdot\mathbf{d}}\sum_{j=1}^{N} e^{-i\mathbf{k}\cdot\mathbf{r}^q_j} = e^{-\frac{1}{2}|\mathbf{k}|^2\sigma_G^2} e^{-i\mathbf{k}\cdot\mathbf{d}} d^q_{\mathbf{k}}. \tag{34}$$

Here we note that the magnitude of the Fourier coefficients decays exponentially, so we can specify a cutoff wavevector, $|\mathbf{k}_{\max}| \gg 1/\sigma_G$ above which we do not need to calculate them. The value of the cutoff will determine the numerical accuracy of the calculation. We find

$$\Omega^{pq}(\mathbf{d}) \approx \frac{1}{L^3}\sum_{\mathbf{k}\in\mathbf{K}}^{|\mathbf{k}|<|\mathbf{k}_{\max}|} e^{-|\mathbf{k}|^2\sigma_G^2} d^p_{\mathbf{k}} d^{q\,*}_{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{d}}. \tag{35}$$

This expression is simply the Fourier series representation of $\Omega^{pq}(\mathbf{d})$, so in order to calculate the maximum value of $\Omega^{pq}(\mathbf{d})$ we can perform the fast inverse Fourier transform (FFT) on the coefficients to calculate an array of values of $\Omega^{pq}(\mathbf{d})$. The value of $\mathbf{d}$ that maximises $\Omega^{pq}(\mathbf{d})$ can be found by fitting a quadratic or Gaussian to the points close to the maximum value of the array in the three axes, or by a local maximisation of eq. (35).

B.2.1 Width of Kernel

From eq. (35) we see that the convolution of a second Gaussian kernel of width σ_{G1} with $\Omega^{pq}(\mathbf{d})$ simply corresponds to the selection of a larger width $\sigma_{G2} = \sqrt{\sigma_G^2 + 2\sigma_{G1}^2}$ for the original Gaussian kernel.

B.2.2 Multiple Species

This method generalises easily to alignment of multicomponent systems, where we can calculate the overlap integral separately for each element, and maximise the sum of the integrals.

B.2.3 Algorithmic Complexity

The efficiency is primarily determined by the number of $\mathbf{k}$ values that need to be computed for the Fourier series representation of $\Omega^{pq}(\mathbf{d})$ to converge. The number of $\mathbf{k}$ values is proportional to the ratio L/σ_G, while $\sigma_G \propto r_{\text{sep}}$, the minimum atomic separation. If we assume that the atomic density is approximately uniform then $r_{\text{sep}} \propto (L/N)^{1/3}$, so the total number of $\mathbf{k}$ values will be proportional to $(L/\sigma_G)^3 \propto N$. Hence calculating the Fourier coefficients will be $O(N^2)$. The FFT will be $O(N \log N)$, so the total complexity of finding the optimal displacement will be $O(N^2)$. Solving the assignment problem to find the correct permutation is also $O(N^2)$, so the overall complexity of the alignment is still $O(N^2)$.

Much of the computational cost is associated with the calculation of the Fourier coefficients in eqs. (33) and (34). When aligning a large database of structures these coefficients can be precalculated, providing a significant performance improvement.

B.2.4 Limitations

This algorithm will fail to find the global RMSD when the difference between the structures is dominated by pairs of atoms that are a large distance apart, as the contribution of these pairs to the overlap integral is small and so will not be optimised. In this case it is possible that the RMSD is not a particularly useful measure of similarity, and so other methods for comparing and aligning structures may be more relevant.

B.3 Minimising RMSD for Clusters

We can perform a very similar analysis for isolated clusters of atoms. For structures p and q, with atomic coordinates $\mathbf{R}^p$ and $\mathbf{R}^q$, and centroids already shifted to the origin, we seek

$$\mathrm{RMSD}(p,q) = \frac{1}{\sqrt{N}} \min_{\alpha,\beta,\gamma,\mathbf{P}} |\mathbf{R}^p - \mathbf{P}\mathbf{R}^q\mathbf{M}(\alpha,\beta,\gamma)^\top|, \tag{36}$$

where $\mathbf{M}$ is the block diagonal coordinate rotation matrix containing N copies of $\mathbf{m}$,

$$\mathbf{m}(\alpha,\beta,\gamma) = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{37}$$

$\mathbf{m}(\alpha,\beta,\gamma)$ is a rotation matrix parameterised by the Euler angles, α, β and γ, representing three successive rotations around the z, y and then the z axis. We can redefine the overlap integral in eq. (7) for rotations:

$$\Omega^{pq}(\alpha,\beta,\gamma) = \iiint_{\infty}^{-\infty} \rho_p(\mathbf{r})\rho_q(\mathbf{m}^T(\alpha,\beta,\gamma)\mathbf{r})\mathrm{d}\mathbf{r} = (\pi\sigma_G^2)^{3/2} \sum_{j=1}^{N}\sum_{j'=1}^{N} \exp\left(-\frac{|\mathbf{r}_j^p - \mathbf{m}(\alpha,\beta,\gamma)\mathbf{r}_{j'}^q|^2}{\sigma_G^2}\right), \tag{38}$$

and similarly, as in eq. (30), for systems where every pair of aligned atoms is separated by much less than σ_G and if $(\alpha_\mathrm{m}, \beta_\mathrm{m}, \gamma_\mathrm{m}) = \arg\max \Omega^{pq}(\alpha, \beta, \gamma)$

$$\Omega^{pq}(\alpha_\mathrm{m}, \beta_\mathrm{m}, \gamma_\mathrm{m}) \approx N(\pi\sigma_G^2)^{3/2} - \pi^{3/2}\sigma_G^{-1/2}\sqrt{N}\,\mathrm{RMSD}(p,q). \tag{39}$$

To evaluate $\Omega^{pq}(\alpha, \beta, \gamma)$ efficiently we follow the method developed to calculate SOAP similarity kernels by Bartók et al.[41] and De et al.[14] based on expanding Gaussian functions by a modified form of the Rayleigh expansion,[68]

$$\exp\left(-\frac{|\mathbf{r}-\mathbf{r}_j|^2}{2\sigma_G^2}\right) = 4\pi\exp\left(-\frac{r^2+r_j^2}{2\sigma_G^2}\right)\sum_{l=0}^{\infty}\sum_{m=-l}^{l} i_l\left(\frac{rr_j}{\sigma_G^2}\right) Y_l^m(\hat{\mathbf{r}})Y_l^m(\hat{\mathbf{r}}_j)^*, \tag{40}$$

where $r = |\mathbf{r}|$ and $\hat{\mathbf{r}} = \mathbf{r}/r$. $Y_l^m(\hat{\mathbf{r}})$ is the value of the spherical harmonic with degree l and order m evaluated at a point on the unit sphere, $\hat{\mathbf{r}}$. $i_l(r)$ are modified spherical Bessel functions of the first kind. Using this relationship we can express the two densities as

$$\begin{aligned}
\rho(\mathbf{m}(\alpha,\beta,\gamma)^T\mathbf{r})_q &= \sum_j\sum_{l=0}^{\infty}\sum_{m,m'=-l}^{l} 4\pi\exp\left(-\frac{r^2+r_j^{q2}}{2\sigma_G^2}\right) i_l\left(\frac{rr_j^q}{\sigma_G^2}\right) D^l_{mm'}(\alpha,\beta,\gamma)Y_l^m(\hat{\mathbf{r}})Y_l^m(\hat{\mathbf{r}}_j^q)^* \\
\rho(\mathbf{r})_p &= \sum_j\sum_{l=0}^{\infty}\sum_{m=-l}^{l} 4\pi\exp\left(-\frac{r^2+r_j^{p2}}{2\sigma_G^2}\right) i_l\left(\frac{rr_j^p}{\sigma_G^2}\right) Y_l^m(\hat{\mathbf{r}})Y_l^m(\hat{\mathbf{r}}_j^p)^*,
\end{aligned} \tag{41}$$

where $D^l_{mm'}(\alpha, \beta, \gamma)$ are the coefficients of the Wigner-D matrix that transforms the coefficients of the spherical harmonics by a rotation of α, β, γ, $\sum_{m'} D^l_{mm'}Y_l^{m'}(\hat{\mathbf{r}}) = Y_l^m(\mathbf{m}\hat{\mathbf{r}})$. For brevity we have dropped the arguments, so $D^l_{mm'} \equiv D^l_{mm'}(\alpha,\beta,\gamma)$ and $\mathbf{m} \equiv \mathbf{m}(\alpha,\beta,\gamma)$, and abbreviated the sums, as $\sum_{l,m}\{\} \equiv \sum_{l=0}^{\infty}\sum_{m=-l}^{l}\{\}$. Substituting eq. (41) into eq. (38)

$$\begin{aligned}
\Omega^{pq}(\alpha,\beta,\gamma) = &\sum_{j,j'}\sum_{l,m}\sum_{l',m',m''}(4\pi)^2 Y_l^m(\hat{\mathbf{r}}_j^p)D^{l'}_{m'm''}Y_{l'}^{m''}(\hat{\mathbf{r}}_{j'}^q)^* \exp\left(-\frac{r_j^{p2}+r_j^{q2}}{2\sigma_G^2}\right)\times \\
&\int_0^{\infty}\exp\left(-\frac{r^2}{\sigma_G^2}\right) i_l\left(\frac{rr_j^p}{\sigma_G^2}\right) i_l\left(\frac{rr_{j'}^q}{\sigma_G^2}\right) r^2\,\mathrm{d}r \int Y_l^m(\hat{\mathbf{r}})^* Y_{l'}^{m''}(\hat{\mathbf{r}})\,\mathrm{d}\hat{\mathbf{r}}.
\end{aligned} \tag{42}$$

The integrals can be evaluated analytically,

$$\int_0^\infty \exp\left(-\frac{r^2}{\sigma_G^2}\right) i_l\left(\frac{r r_j^p}{\sigma_G^2}\right) i_l\left(\frac{r r_{j'}^q}{\sigma_G^2}\right) r^2 \, \mathrm{d}r = \frac{\sqrt{\pi}\sigma_G^3}{4} i_l\left(\frac{r_j^p r_{j'}^q}{2\sigma_G^2}\right) \exp\left(\frac{r_j^{p\,2} + r_{j'}^{q\,2}}{4\sigma_G^2}\right), \tag{43}$$

$$\int Y_l^m(\hat{\mathbf{r}})^* Y_{l'}^{m''}(\hat{\mathbf{r}}) \, \mathrm{d}\hat{\mathbf{r}} = \delta_{ll'}\delta_{mm''}. \tag{44}$$

Hence

$$\Omega^{pq}(\alpha,\beta,\gamma) = \sum_{j,j'} \sum_{l,m,m'} 4\pi^{5/2}\sigma_G^3 \, Y_l^m(\hat{\mathbf{r}}_j^p) Y_l^{m'}(\hat{\mathbf{r}}_{j'}^q)^* \exp\left(-\frac{r_j^{p\,2} + r_{j'}^{q\,2}}{4\sigma_G^2}\right) i_l\left(\frac{r_j^p r_{j'}^q}{2\sigma_G^2}\right) D_{mm'}^l. \tag{45}$$

Now we can calculate the Fourier coefficients of the overlap integral as,

$$\Omega^{pq}(\alpha,\beta,\gamma) = \sum_{l=0}^{l_{\max}} \sum_{m,m'=-l}^{l} I_{mm'}^l D_{mm'}^l(\alpha,\beta,\gamma), \tag{46}$$

$$\text{where, } I_{mm'}^l = \sum_{j,j'} 4\pi^{5/2}\sigma_G^3 \, Y_l^m(\hat{\mathbf{r}}_j^p) Y_l^{m'}(\hat{\mathbf{r}}_{j'}^q)^* \exp\left(-\frac{r_j^{p\,2} + r_{j'}^{q\,2}}{4\sigma_G^2}\right) i_l\left(\frac{r_j^p r_{j'}^q}{2\sigma_G^2}\right). \tag{47}$$

To evaluate the integral numerically we truncate the the sum at a maximum angular momentum degree, $l_{\max}$. To find the maximum value of Ω^{pq} we can use SOFT to perform a $SO(3)$ Fourier synthesis on $I_{mm'}^l$, which can be achieved in $O(\Delta_\theta^3 \log^2 \Delta_\theta)$ operations, where $2\Delta_\theta = l_{\max}$ is the angular resolution or bandwidth of SOFT.[31] Most implementations of SOFT (including ours) have $O(\Delta_\theta^4)$ computational complexity.

Calculating the full sum over all j and j' is computationally expensive so the number terms required can be reduced by omitting contributions where $|r_j^p - r_{j'}^q| \gg \sigma_G$. Assuming a uniform density of points this observation means the number of terms in the sum will reduce from N^2 to $N^{5/3}$. Hence, calculating the Fourier coefficients requires $O(N^{5/3} l_{\max}^3)$ operations.

The result given by this Fourier synthesis can be refined by performing a local minimisation of eq. (46), as the gradients of $D_{mm'}^l(\alpha,\beta,\gamma)$ can be calculated analytically or by fitting a set of Gaussian peaks to the output data, and the location of these peaks can be used as an initial starting point for the PERMDIST algorithm (see section 1.1.2).

B.3.1 Harmonic Basis

The calculation of the cross terms in eq. (43) makes the overlap method more expensive, and as a result, it is harder to evaluate alternative metrics. To improve efficiency we can project eq. (41) onto an orthogonal radial basis, which we can generate from the isotropic three-dimensional quantum harmonic oscillator (referred to here as the harmonic basis). Expressing eq. (41) in the harmonic basis we obtain,

$$\rho_p(\mathbf{r}) = \sum_{n,l,m} c^p_{nlm} N_{nl} r^l \exp\left(-\frac{r^2}{2r_0^2}\right) L_n^{l+1/2}\left(\frac{r^2}{r_0^2}\right) Y_l^m(\hat{\mathbf{r}}) = \sum_{n,l,m} c^p_{nlm} g_{nl}(r) Y_l^m(\hat{\mathbf{r}}), \tag{48}$$

where $L_n^m(r)$, are generalised Laguerre polynomials and

$$N_{nl} = \sqrt{\frac{2n!}{r_0^{2l+3}\Gamma(3/2+n+l)}} \tag{49}$$

is the normalisation constant, such that,

$$\iiint g_{nl}(r) Y_l^m(\hat{\mathbf{r}})^* g_{n'l'}(r) Y_{l'}^{m'}(\hat{\mathbf{r}})\, \mathrm{d}\mathbf{r} = \delta_{nn'}\delta_{ll'}\delta_{mm'}. \tag{50}$$

The coefficients of eq. (48) can be obtained using eqs. (40) and (50),

$$\begin{aligned} c^p_{nlm} = \int \rho_p(\mathbf{r}) g_{nl}(r) Y_l^m(\hat{\mathbf{r}})\, \mathrm{d}\mathbf{r} &= 4\pi \sum_{j=1}^{N} Y_l^m(\hat{\mathbf{r}}_j^p)^* \int_0^\infty g_{nl}(r) \exp\left(-\frac{r^2+r_j^{p2}}{2\sigma_G^2}\right) i_l\left(\frac{r r_j^p}{\sigma_G^2}\right) r^2\, \mathrm{d}r \\ &= \sum_{j=1}^{N} Y_l^m(\hat{\mathbf{r}}_j^p)^* d_{nl}(r_j^p). \end{aligned} \tag{51}$$

For $n = 0$ we have the following analytic result,

$$d_{0,l} = 4\sigma_G^3 \sqrt{\frac{\pi^3}{r_j^3 \Gamma\left(l+\frac{3}{2}\right)}} \left(\frac{r_0 r_j}{r_0^2+\sigma_G^2}\right)^{l+\frac{3}{2}} \exp\left(-\frac{r_j^2}{2\left(r_0^2+\sigma_G^2\right)}\right). \tag{52}$$

To evaluate eq. (51) for larger values of n we can use the following recurrence relations,

$$nL_n^{l+1/2}(x) = (n+l+1/2)L_{n-1}^{l+1/2}(x) - xL_{n-1}^{l+3/2} \tag{53}$$

$$L_{n-1}^{l+3/2}(x) = L_{n-1}^{l+5/2}(x) - L_{n-2}^{l+5/2}(x) \tag{54}$$

$$i_l(x) = \frac{2l+3}{x} i_{l+1}(x) + i_{l+2}(x). \tag{55}$$

Hence we obtain a recurrence relation for the integral $d_{n,l}(r_j)$ (where for brevity we drop the argument of $d_{n,l}(r_j)$, so $d_{n,l}(r_j) \equiv d_{n,l}$, and noting that $d_{-1,l} = 0$),

$$0 = -\sqrt{\frac{n-1}{n}}\, d_{n-2,l+2} - \sqrt{\frac{n+l+1/2}{n}}\, d_{n-1,l} + \frac{(2l+3)\sigma_G^2}{r_j r_0 \sqrt{n}} d_{n-1,l+1} + \sqrt{\frac{n+l+3/2}{n}}\, d_{n-1,l+2} + d_{n,l}. \tag{56}$$

Unfortunately evaluating the forward recurrence of eq. (56) is numerically unstable for large n and l, and attempting to stabilise the recursion results in ill-conditioned matrices. This problem limits the ratio of system spatial size of the system to the width of the kernel, $\max(r_j) < 10\sigma_G$. Within this regime Fourier coefficients of the overlap integral can be calculated,

$$\begin{aligned}\Omega^{pq}(\alpha,\beta,\gamma) &= \sum_{n,l,m}\sum_{n',l',m'}\sum_{m''} \iiint (c^p_{n,l,m})^* g_{nl}(r) Y_l^m(\hat{\mathbf{r}})^*\, c^p_{n',l',m'} g_{n'l'}(r) D^{l'}_{m''m'} Y_{l'}^{m''}(\hat{\mathbf{r}}) \mathrm{d}\mathbf{r} \\ &= \sum_n (c^p_{n,l,m})^* c^q_{n,l,m'} D^l_{mm'}. \end{aligned} \tag{57}$$

This result can be used to perform an alignment by following the method in appendix B.3. When aligning a large database of structures the algorithm can be made more efficient by precalculating the harmonic basis coefficients. This precalculation will come at a slight cost of accuracy in eq. (57), as only a fixed number of radial basis functions can be considered, whereas the numerical calculation for eq. (47) is exact.

Computational Complexity Calculating eq. (57) requires specifying a cutoff angular momentum order, as discussed in appendix B.3, and a cutoff harmonic basis order, such

that $n \leq n_{\max}$ and $n_{\max} \propto N^{2/3}$. Hence the total complexity associated with calculating the harmonic basis coefficients will be approximately $O(N^{5/3} l_{\max}^3)$.

B.3.2 Spherical Fourier Transforms

Generalising the Fourier transform to spherical coordinates gives an alternate method to obtain the SO(3) Fourier coefficients. For a function $f(\mathbf{r})$, the Fourier transform and Fourier synthesis can be defined,

$$F(\mathbf{k}) = \mathcal{F}[f(\mathbf{r})]_{\mathbf{k}} = \iiint f(\mathbf{r}) \exp(-i\mathbf{k} \cdot \mathbf{r}) \, \mathrm{d}\mathbf{r} \tag{58}$$

$$f(\mathbf{r}) = \mathcal{F}^{-1}[f(\mathbf{k})]_{\mathbf{r}} = \frac{1}{(2\pi)^3} \iiint F(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{r}) \, \mathrm{d}\mathbf{k}. \tag{59}$$

This approach can be generalised to spherical coordinates by expressing the exponential in spherical harmonics,

$$\exp(i\mathbf{k} \cdot \mathbf{r}) = 4\pi \sum_{l,m} i^l j_l(kr) Y_l^m(\hat{\mathbf{r}}) Y_l^m(\hat{\mathbf{k}})^*. \tag{60}$$

where $j_l(kr)$ are spherical Bessel functions of the first kind. We can use Parseval's theorem to evaluate the overlap integral of the two densities,

$$\begin{aligned} \Omega^{pq}(\alpha, \beta, \gamma) &= \iiint_{\infty}^{-\infty} \rho_p(\mathbf{r}) \rho_q(\mathbf{mr})^* \mathrm{d}\mathbf{r} \\ &= \frac{1}{(2\pi)^3} \iiint_{\infty}^{-\infty} \mathcal{F}\left[\rho_p(\mathbf{r})\right]_{\mathbf{k}} \mathcal{F}\left[\rho_q(\mathbf{mr})\right]_{\mathbf{k}}^* \mathrm{d}\mathbf{k} \end{aligned} \tag{61}$$

Using the convolution theorem we can calculate the Fourier transforms,

$$\begin{aligned} C^p(\mathbf{k}) &= \mathcal{F}\left[\rho_p(\mathbf{r})\right]_{\mathbf{k}} \\ &= (2\pi\sigma_G^2)^{3/2} \exp\left(-\frac{k^2\sigma_G^2}{2}\right) \sum_j \exp(-i\mathbf{k} \cdot \mathbf{r}_j^p) \end{aligned}$$

$$= (2\pi\sigma_G^2)^{3/2} \exp\left(-\frac{k^2\sigma_G^2}{2}\right) 4\pi \sum_j \sum_{l,m} (-i)^l j_l(\kappa r_j^p) Y_l^m(\hat{\mathbf{r}}_j^p)^* Y_l^m(\hat{\mathbf{k}}), \tag{62}$$

$$C^q(\mathbf{k}) = \mathcal{F}\left[\rho_q(\mathbf{mr})\right]_{\mathbf{k}}$$
$$= (2\pi\sigma_G^2)^{3/2} \exp\left(-\frac{k^2\sigma_G^2}{2}\right) 4\pi \sum_j \sum_{l,m} \sum_{m'} (-i)^l j_l(k r_j^q) D^l_{m,m'} Y_l^{m'}(\hat{\mathbf{r}}_j^q)^* Y_l^m(\hat{\mathbf{k}}). \tag{63}$$

Instead of evaluating eq. (61) as an integral we can evaluate it as a sum by considering the discrete spherical Fourier transform, where we truncate the integral up to a cut-off radius, r_{cut}, and use the following orthogonality relation,

$$\int_0^{r_{\text{cut}}} j_l\left(\frac{\kappa_{l,n}}{r_{\text{cut}}} r\right) j_l\left(\frac{\kappa_{l,n'}}{r_{\text{cut}}} r\right) r^2 \,\mathrm{d}r = \frac{\pi\, r_{\text{cut}}^3}{4\kappa_{l,n}} j_{l+1}(\kappa_{l,n})^2 \delta_{n'n}, \tag{64}$$

where $\kappa_{l,n}$ is the nth root of j_l, so $j_l(\kappa_{l,n}) = 0$, to transform eq. (59) into a sum (the spherical analogue of a discrete Hankel transform,[69])

$$f(\mathbf{r}) = \sum_{l,m} \sum_{n=1}^{\infty} F_{l,n}^m \sqrt{\frac{2\kappa_{l,n}}{r_{\text{cut}}^3 j_{l+1}(\kappa_{l,n})^2}}\, 4\pi i^l j_l\left(\frac{\kappa_{l,n}}{r_{\text{cut}}} r\right) Y_l^m(\hat{\mathbf{r}}), \tag{65}$$

$$F_{l,n}^m = \frac{1}{(2\pi)^3} \sqrt{\frac{2\kappa_{l,n}}{r_{\text{cut}}^3 j_{l+1}(\kappa_{l,n})^2}} \iiint_{|\mathbf{r}|<r_{\text{cut}}} f(\mathbf{r}) 4\pi (-i)^l j_l\left(\frac{\kappa_{l,n}}{r_{\text{cut}}} r\right) Y_l^m(\hat{\mathbf{r}})^* \,\mathrm{d}\mathbf{r}. \tag{66}$$

If we assume that $\rho_p(\mathbf{r}) = \rho_q(\mathbf{r}) = 0$ when $|\mathbf{r}| \geq r_{\text{cut}}$, then by inspection of eqs. (62) and (63) we can express the density functions as,

$$\rho_p(\mathbf{r}) = \sum_{l,m} \sum_{n=1}^{\infty} C_{l,n}^{p,m} \sqrt{\frac{2\kappa_{l,n}}{r_{\text{cut}}^3 j_{l+1}(\kappa_{l,n})^2}}\, 4\pi i^l j_l\left(\frac{\kappa_{l,n}}{r_{\text{cut}}} r\right) Y_l^m(\hat{\mathbf{r}}), \tag{67}$$

$$C_{l,n}^{p,m} = (2\sigma_G^2)^{3/2} \sum_{j=1}^{N} \sqrt{\frac{2\kappa_{l,n}}{r_{\text{cut}}^3 j_{l+1}(\kappa_{l,n})^2}} \exp\left(-\frac{\kappa_{l,n}^2\sigma_G^2}{2r_{\text{cut}}^2}\right) j_l\left(\frac{\kappa_{l,n}}{r_{\text{cut}}} r_j^p\right) Y_l^m(\hat{\mathbf{r}}_j^p)^*, \tag{68}$$

$$\rho_q(\mathbf{mr}) = \sum_{l,m} \sum_{m'} \sum_{n=1}^{\infty} D^l_{m,m'} C_{l,n}^{p,m} \sqrt{\frac{2\kappa_{l,n}}{r_{\text{cut}}^3 j_{l+1}(\kappa_{l,n})^2}}\, 4\pi i^l j_l\left(\frac{\kappa_{l,n}}{r_{\text{cut}}} r\right) Y_l^{m'}(\hat{\mathbf{r}}), \tag{69}$$

$$C_{l,n}^{p,m} = (2\sigma_G^2)^{3/2} \sum_{j=1}^{N} \sqrt{\frac{2\kappa_{l,n}}{r_{\text{cut}}^3 j_{l+1}(\kappa_{l,n})^2}} \exp\left(-\frac{\kappa_{l,n}^2\sigma_G^2}{2r_{\text{cut}}^2}\right) j_l\left(\frac{\kappa_{l,n}}{r_{\text{cut}}} r_j^q\right) Y_l^m(\hat{\mathbf{r}}_j^q)^*. \tag{70}$$

We can obtain an expression for the overlap integral,

$$\Omega^{pq}(\alpha,\beta,\gamma) = \sum_{l,m,m'} D^l_{m,m'} \frac{1}{(2\pi)^3} \sum_n^{\infty} C^{p,m*}_{l,n} C^{q,m'}_{l,n}. \tag{71}$$

To evaluate the overlap integral we need to specify a cut-off order, n^l_{cut}, such that

$$\kappa_{l,n^l_{\text{cut}}} \gg \frac{r_{\text{cut}}}{\sigma_G}, \tag{72}$$

as the zeros of the spherical Bessel function are approximately uniformly distributed, $\kappa_{l,n^l_{\text{cut}}} \propto N^{1/3}$. The SO(3) Fourier coefficients can be calculated,

$$I_{l,m,m'} = \frac{1}{(2\pi)^3} \sum_n^{n^l_{\text{cut}}} C^{p,m*}_{l,n} C^{q,m'}_{l,n}, \tag{73}$$

which will require $O(l^3_{\text{max}} N^{1/3})$ operations, while calculating the spherical Fourier coefficients will require $O(l^2_{\text{max}} N^{4/3})$ operations.

B.4 Including Multiple Species

The above methods assume that there is only one type of atomic species present; it is straightforward to extend the FASTOVERLAP methods to apply to systems with multiple different types of atomic species. In the multiple species case, the Fourier or Harmonic coefficients for each species can be calculated independently. The Fourier coefficients for the species overlap integral can be calculated separately in turn. The coefficients for the total overlap integral are equal to the sum of the species overlap Fourier coefficients. In the case of the method described in appendix B.3 the method is modified by only summing over the indexes of identical atomic species in eq. (47).

References

(1) Verma, J.; Khedkar, V.; Coutinho, E. 3D-QSAR in Drug Design - A Review. *Curr. Top. Med. Chem.* **2010**, *10*, 95.

(2) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big Data Meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087–2096.

(3) Huo, H.; Rupp, M. Unified Representation for Machine Learning of Molecules and Crystals. *arXiv preprint arXiv:1704.06439* **2017**, Accessed: 2017-08-22.

(4) Rosenbrock, C. W.; Homer, E. R.; Csányi, G.; Hart, G. L. W. Discovering the Building Blocks of Atomic Systems using Machine Learning. *arXiv preprint arXiv:1703.06236* **2017**, Accessed: 2017-08-22.

(5) Bartók, A. P.; Csányi, G. Gaussian approximation potentials: A brief tutorial introduction. *Int. J. Quantum Chem.* **2015**, *115*, 1051–1057.

(6) Wales, D. J. Discrete path sampling. *Mol. Phys.* **2002**, *100*, 3285.

(7) Wales, D. J. Some further applications of discrete path sampling to cluster isomerization. *Mol. Phys.* **2004**, *102*, 891.

(8) Wales, D. J. Energy Landscapes: Calculating Pathways and Rates. *Int. Rev. Phys. Chem.* **2006**, *25*, 237.

(9) Carr, J. M.; Trygubenko, S. A.; Wales, D. J. Finding pathways between distant local minima (7 pages). *J. Chem. Phys.* **2005**, *122*, 234903.

(10) Wales, D. J.; Carr, J. M. Quasi-Continuous Interpolation Scheme for Pathways between Distant Configurations. *J. Chem. Theory Comput.* **2012**, *8*, 5020.

(11) Bauer, M. S.; Strodel, B.; Fejer, S. N.; Koslover, E. F.; Wales, D. J. Interpolation schemes for peptide rearrangements. *J. Chem. Phys.* **2010**, *132*, 054101.

(12) Jonker, R.; Volgenant, A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* **1987**, *38*, 325.

(13) Sadeghi, A.; Ghasemi, S. A.; Schaefer, B.; Mohr, S.; Lill, M. A.; Goedecker, S. Metrics for measuring distances in configuration spaces. *J. Chem. Phys.* **2013**, *139*, 184118.

(14) De, S.; Bartók, A. P.; Csányi, G.; Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.* **2016**, *18*, 13754.

(15) Salvi, J.; Matabosch, C.; Fofi, D.; Forest, J. A review of recent range image registration methods with accuracy evaluation. *Image Vis. Comput.* **2007**, *25*, 578.

(16) Hill, D. L. G.; Hawkes, D. J.; Crossman, J. E.; Gleeson, M. J.; Cox, T. C. S.; Bracey, E. E. C. M. L.; Strong, A. J.; Graves, P. Registration of MR and CT images for skull base surgery using point-like anatomical features. *Br. J. Radiol.* **1991**, *64*, 1030.

(17) Fitzgibbon, A. W. Robust registration of 2D and 3D point sets. *Image Vis. Comput.* **2003**, *21*, 1145.

(18) Besl, P.; McKay, N. D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239.

(19) Tsin, Y.; Kanade, T. In *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III*; Pajdla, T., Matas, J., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2004; p 558.

(20) Makadia, A.; Patterson, A. I.; Daniilidis, K. Fully Automatic Registration of 3D Point Clouds. *2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Vol. 1* **2006**, *1*, 1297.

(21) Yang, J.; Li, H.; Jia, Y. Go-ICP: Solving 3D registration efficiently and globally optimally. *Proc. IEEE Int. Conf. Comput. Vis.* **2013**, 1457.

(22) Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2241.

(23) Wales, D. J. *Energy Landscapes*; Cambridge University Press: Cambridge, 2003; pp 165–170.

(24) Kuhn, H. W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83.

(25) Coutsias, E. A.; Seok, C.; Dill, K. A. Using quaternions to calculate RMSD. *J. Comput. Chem.* **2004**, *25*, 1849.

(26) Kabsch, W. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. Sect. A* **1978**, *34*, 827.

(27) Schaefer, B.; Goedecker, S. Computationally efficient characterization of potential energy surfaces based on fingerprint distances. *J. Chem. Phys.* **2016**, *145*, 034101.

(28) Granger, S.; Pennec, X. Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration. *Eur. Conf. Comput. Vis. (ECCV 2002), Vol. 2353 LNCS* **2002**, 418.

(29) Li, H.; Hartley, R. The 3D-3D Registration Problem Revisited. Proc. IEEE Int. Conf. Comput. Vis. 2007; p 1.

(30) Horn, B. K. P. Extended Gaussian Images. *Proc. IEEE* **1984**, *72*, 1671.

(31) Kostelec, P. J.; Rockmore, D. N. FFTs on the rotation group. *J. Fourier Anal. Appl.* **2008**, *14*, 145.

(32) Comin, M.; Guerra, C.; Dellaert, F. Binding balls: fast detection of binding sites using a property of spherical Fourier transform. *J. Comput. Biol.* **2009**, *16*, 1577.

(33) Padhorny, D.; Kazennov, A.; Zerbe, B. S.; Porter, K. A.; Xia, B.; Mottarella, S. E.; Kholodov, Y.; Ritchie, D. W.; Vajda, S.; Kozakov, D. Protein-protein docking by fast generalized Fourier transforms on 5D rotational manifolds. *Proc. Natl. Acad. Sci.* **2016**, *113*, E4286–E4293.

(34) Hong, E.-J.; Lee, K.-H.; Wenzel, W. RMSD computation for clusters of identical particles. Proc. 4th WSEAS Conf. Math. Biol. 2008; p 46.

(35) Wales, D. J. GMIN: A program for basin-hopping global optimisation, basin-sampling, and parallel tempering. http://www-wales.ch.cam.ac.uk/software.html, Accessed: 2017-08-22.

(36) Wales, D. J. OPTIM: A program for geometry optimisation and pathway calculations. http://www-wales.ch.cam.ac.uk/software.html, Accessed: 2017-08-22.

(37) de Souza, V. K.; Wales, D. J. The potential energy landscape for crystallisation of a Lennard-Jones fluid. *J. Stat. Mech.* **2016**, *2016*, 074001.

(38) Hundt, R. KPLOT: A Program for Plotting and Investigation of Crystal Structures. University of Bonn, Germany.

(39) Hundt, R.; Schön, J. C.; Jansen, M. CMPZ - An algorithm for the efficient comparison of periodic structures. *J. Appl. Crystallogr.* **2006**, *39*, 6.

(40) Hundt, R.; Schön, J. C.; Neelamraju, S.; Zagorac, J.; Jansen, M. CCL: An algorithm for the efficient comparison of clusters. *J. Appl. Crystallogr.* **2013**, *46*, 587.

(41) Bartók, A. P.; Kondor, R.; Csányi, G. On representing chemical environments. *Phys. Rev. B* **2013**, *87*, 184115.

(42) Ferré, G.; Maillet, J. B.; Stoltz, G. Permutation-invariant distance between atomic configurations. *J. Chem. Phys.* **2015**, *143*, 104114.

(43) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, 1.

(44) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **2010**, *104*, 1.

(45) Behler, J. Perspective: Machine learning potentials for atomistic simulations. *J. Chem. Phys.* **2016**, *145*, 170901.

(46) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, *108*, 058301.

(47) Zhu, L.; Amsler, M.; Fuhrer, T.; Schaefer, B.; Faraji, S.; Rostami, S.; Ghasemi, S. A.; Sadeghi, A.; Grauzinyte, M.; Wolverton, C.; Goedecker, S. A fingerprint based metric for measuring similarities of crystalline structures. *J. Chem. Phys.* **2016**, *144*, 034203.

(48) Ramírez, M.; Rogan, J.; Valdivia, J. A.; Varas, A.; Kiwi, M. Diversity characterization of binary clusters by means of a generalized distance. *Zeitschrift fur Phys. Chemie* **2016**, *230*, 977.

(49) Steinhardt, P. J.; Nelson, D. R.; Ronchetti, M. Bond-orientational order in liquids and glasses. *Phys. Rev. B* **1983**, *28*, 784.

(50) Kondor, R. A complete set of rotationally and translationally invariant features for images. *CoRR* **2007**, *abs/cs/0701127*.

(51) Gelbrich, T.; Threlfall, T. L.; Hursthouse, M. B. XPac dissimilarity parameters as quantitative descriptors of isostructurality: the case of fourteen 4,5[prime or minute]-substituted benzenesulfonamido-2-pyridines obtained by substituent interchange involving CF3/I/Br/Cl/F/Me/H. *CrystEngComm* **2012**, *14*, 5454.

(52) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106.

(53) Pietrucci, F.; Andreoni, W. Graph theory meets ab initio molecular dynamics: Atomic structures and transformations at the nanoscale. *Phys. Rev. Lett.* **2011**, *107*, 1.

(54) Barker, J.; Bulin, J.; Hamaekers, J.; Mathias, S. Localized Coulomb Descriptors for the Gaussian Approximation Potential. **2016**, 1.

(55) Valle, M.; Oganov, A. R. Crystal fingerprint space - A novel paradigm for studying crystal-structure sets. *Acta Crystallogr. Sect. A Found. Crystallogr.* **2010**, *66*, 507.

(56) Saberi Fathi, S. M.; White, D. T.; Tuszynski, J. A. Geometrical comparison of two protein structures using Wigner-D functions. *Proteins Struct. Funct. Bioinforma.* **2014**, *82*, 2756.

(57) Stoddard, S. D.; Ford, J. Numerical experiments on the stochastic behavior of a Lennard-Jones system. *Phys. Rev. A* **1973**, *8*, 1504.

(58) Kob, W.; Andersen, H. C. Testing mode-coupling theory for a supercooled binary Lennard-Jones mixture: The van Hove correlation function. *Phys. Rev. E* **1995**, *51*, 4626.

(59) PELE: Python Energy Landscape Explorer, https://github.com/pele-python/pele. `https://github.com/pele-python/pele`, Accessed: 2017-08-22.

(60) Wales, D. J.; Doye, J. P. K. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 Atoms. *J. Phys. Chem. A* **1997**, *101*, 5111.

(61) Li, Z.; Scheraga, H. A. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *Proc. Natl. Acad. Sci. USA* **1987**, *84*, 6611.

(62) Li, Z.; Scheraga, H. A. Structure and free energy of complex thermodynamic systems. *J. Mol. Struct.* **1988**, *179*, 333.

(63) Trygubenko, S. A.; Wales, D. J. Analysis of Cooperativity and Localization for Atomic Rearrangements. *J. Chem. Phys.* **2004**, *121*, 6689.

(64) Len, C. A.; Mass, J.-C.; Rivest, L.-P. A statistical model for random rotations. *J. Multivar. Anal.* **2006**, *97*, 412.

(65) Rosato, V.; Guillope, M.; Legrand, B. Thermodynamical and structural properties of fcc transition metals using a simple tight-binding model. *Phil. Mag. A* **1989**, *59*, 321.

(66) Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* **2007**, *9*, 90.

(67) Freidman, J. H.; Bentley, J. L.; Finkel, R. A. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. Math. Softw.* **1977**, *3*, 209.

(68) Kaufmann, K.; Baumeister, W. Single-centre expansion of Gaussian basis functions and the angular decomposition of their overlap integrals. *J. Phys. B At. Mol. Opt. Phys.* **1989**, *22*, 1.

(69) Baddour, N.; Chouinard, U. Theory and operational rules for the discrete Hankel transform. *J. Opt. Soc. Am. A* **2015**, *32*, 611.